Universitat
Autònoma
de Barcelona

2443: Music Identification

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Ignasi Vila Tudela
i dirigit per
Mireia Bellot Garcia
Bellaterra, 17 de Setembre de 2010

Agraïments

Vull en primer lloc donar les gràcies a la meva família pel seu suport i motivació durant tots aquests anys, sense els quals difícilment hagués arribat on sóc ara.

També als companys de la universitat, ja que han estat una font de coneixement i de diversió incomparable, i que segur, a partir d'ara, trobaré molt a faltar.

I com no, al tots els meus amics, que han estat sempre allà, independentment de si jo hi era o no.

També a en Xavier, com a coordinador del projecte, per saber està allà en tot moment i per la seva desinteressada col·laboració. I finalment i en especial a la meva directora, la Mireia, pels seus encertats consells, i per la seva dedicació en els moments clau.

Índex de continguts

1 Introducció

1.1 Motivacions

És un fet que bona part de les lletres de la música que escoltem habitualment es troba en llengües que, en major o menor mesura, desconeixem. Això implica perdre una gran quantitat del missatge, i per tant, de l'obra.

Solucionar aquest problema seria tan fàcil com crear un sistema que fos capaç de rebre una mostra de so, identificar la cançó a la qual pertany així com el temps en què es troba, i retornar les lletres a temps real, com si fos un karaoke.

Aquest sistema estaria format per dues parts principals, la primera seria la responsable d'identificar la cançó així com la situació del so dins de la cançó, i la segona, haura d'encarregar-se d'obtenir les lletres. La segona part és força trivial, una base de dades amb les lletres de les cançons, juntament amb el temps en el qual es canta cada vers. Mentre que la primera part és força més complicada.

1.2 Objectiu

L'objectiu del projecte és investigar quins algoritmes existeixen actualment que resolen el problema d'identificar una cançó a partir d'una mostra de so, i fer un estudi comparatiu d'alguns d'ells per a finalment descobrir quin és el que ofereix millors resultats.

Per a fer-ho, en primer lloc, estudiarem la literatura sobre el tema, per a posteriorment seleccionar els que ens semblin més prometedors, implementar-los i veure quins són els que ofereixen millors resultats.

1.3 Organització de la memòria

Trobant-nos en el penúltim apartat del Capítol 1, en el **Capítol 2 (Fonaments teòrics)**, hem fet una petita introducció teòrica dels conceptes amb que treballarem posteriorment. Constarà d'una introducció al processament informàtic del so, així com els principals elements matemàtics relacionats amb aquest.

En el **Capítol 3 (Estat de l'art)**, farem un profund anàlisi de l'estat de l'art dels algoritmes existents actualment.

En primer lloc veurem els passos en què generalment consisteixen, i posteriorment en farem una explicació detallada de cadascun d'ells.

Finalment explicarem en profunditat el funcionament dels algoritmes que hem implementat.

En el **Capítol 4 (Disseny)**, s'explica detalladament la implementació dels algoritmes que hem triat per a sotmetre'ls al joc de proves.

En el **Capítol 5 (Resultats i Conclusions)** expliquem com s'ha dissenyat l'entorn de proves i perquè, per a finalment mostrar els resultats que ha tingut cadascun dels algoritmes, així com perquè creiem que han estat deguts.

Finalment, al **Capítol 6 (Ampliacions i millores)**, expliquem els passos futurs.

A més a més, els capítols posteriors hi trobem la bibliografia, els resums (en català, castellà i anglès), l'índex de figures i l'índex de taules.

1.4 Planificació

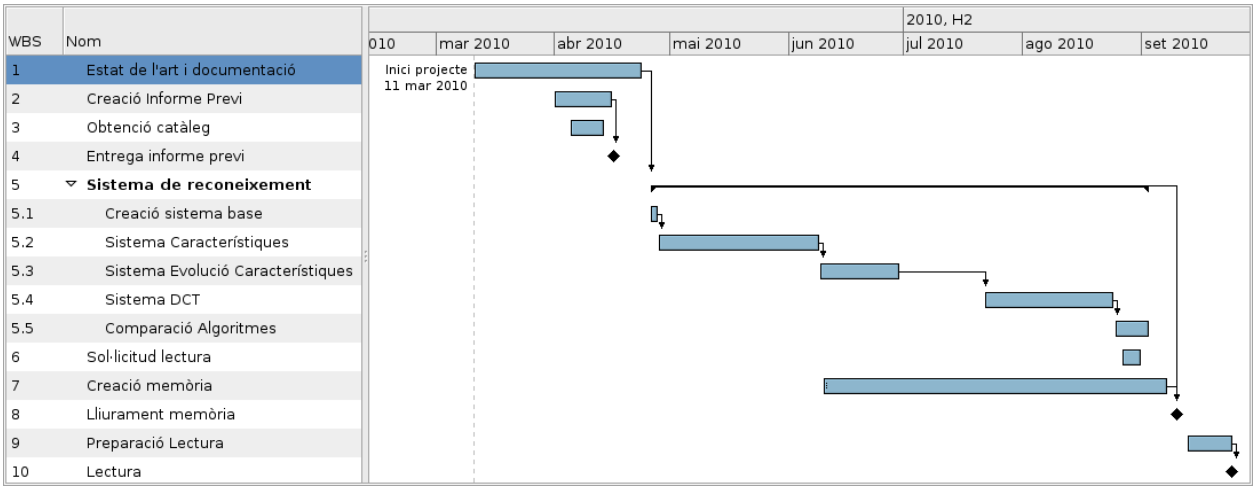


Figura 1.4.1 : Planificació

2 Fonaments Teòrics

2.1 El So

Anomenem so a les ones electromagnètiques que es propaguen per un medi (líquid, sòlid o gasós), que tenen una longitud d'ona determinada i que tenen una intensitat prou elevada com per estimular el òrgans de l'oïda.

El rang de freqüències que activen el sistema auditiu humà, es troba entre 12 i 20.000 Hz, tot i que el límit superior va disminuint al llarg dels anys (recordem la polèmica que va sorgir fa uns anys per uns emissors de sons d'alta freqüència que tenien la finalitat de molestar només als més joves, per tal de que no s'hi apropessin).

Altres espècies tenen òrgans auditius adaptats a altres rangs, com per exemple els gossos, que poden notar sons de freqüències més elevades, o les balenes, que poden notar sons de freqüències molt baixes, cosa que els permet comunicar-se a través de grans distàncies.

Figura 2.1.1: Mostra d'una ona sonora

L'acústica és una ciència interdisciplinar que s'ocupa del so i de la seva transmissió, i a la pràctica s'ocupa d'assegurar que la transmissió dels sons a través de diferents medis és adequada, per exemple a l'hora de dissenyar un teatre, o construir un micròfon.

Els sons, en funció de com els percebem, poden ser classificats en tres atributs principals [1]:

- **Pitch** (o Altura) : Ens indica quina és la freqüència fonamental que percebem. En funció del pitch, direm que un so és greu o agut.
- **Intensita**t : Ens informa de l'amplada, l'energia, de la vibració. També conegut com el volum. Ens permet dir si un so té un volum alt o baix.
- **Timbre** : Ens permet diferenciar entre sons amb el mateix pitch i intensitat. És un atribut complex, que sovint és anomenat com textura o color del so. Es defineix en funció de la forma que té el seu espectre, independentment del pitch o de la intensitat del so.

La percepció del pitch i de la intensitat no és lineal, per tant, un increment del 100% en qualsevol d'aquestes característiques, no implicarà un percepció equivalent per un oient.

Per a modelar de manera més precisa la percepció humana, existeixen escales alternatives a la lineal, que pretenen assimilar la variació del pitch a la percepció humana. Algunes d'aquestes escales, són per exemple, la de Mel o la de Bark.

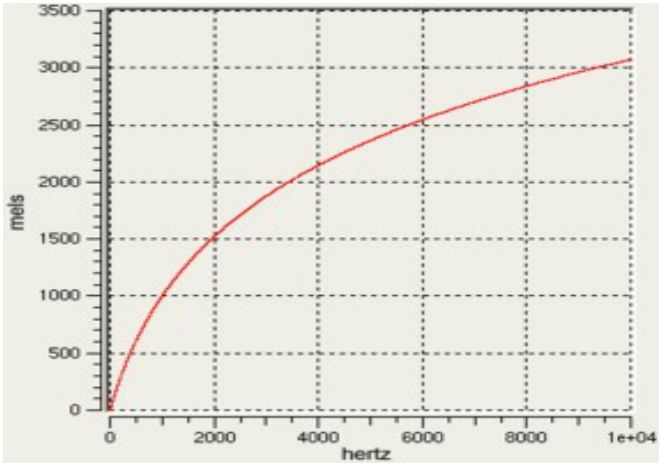


Figura 2.1.2: Relació entre l'escala lineal i la de Mel

El timbre per la seva banda, ens permet diferenciar per exemple entre diferents instruments, o també, si un mateix instrument està sonant en una habitació, o en un teatre, així com moltes altres característiques. Dit això, no és sorprenent que no el definim pel que és, sinó pel que no és.

2.2 La Música

Anomenem música, a la composició de sons i silencis al llarg del temps, amb una finalitat artística.

Inicialment els instruments utilitzats per a fer música eren flautes rudimentàries que produïen sons de diferents freqüències, però amb el pas del temps, han aparegut nous instruments, que han permès als compositors expressar-se d'una manera molt més elaborada.

Per exemple, avui en dia trobem instruments que no generen només sons elementals (d'una freqüència determinada), sinó que generen acords, és a dir, conjunts de sons de diferents freqüències, com per exemple les guitarres i els pianos. De totes maneres, encara existeixen diversos instruments, que generen sons d'una sola freqüència, com per exemple la majoria dels de vent, amb l'excepció, per exemple, dels acordions.

2.3 Característiques del llenguatge musical

Totes aquestes possibilitats fan que s'hagi creat tota una terminologia que descriu les diferents interaccions que es poden produir entre els diferents instruments. Podem parlar del llenguatge musical, el qual ens permet caracteritzar diferents tipus de música en funció de diferents dimensions.

Els que més ens interessen a l'hora de classificar un so, són els següents [1]:

- **Timbre**: Ens dóna una idea de quin és l'origen d'un so.
- **Orquestració**: Ens indica quins són els instruments encarregats de tocar cada veu, acord o cop d'una composició.
- **Acústica**: Ens indica quin és el medi mitjançant el qual s'està transmetent el so.
- **Ritme**: Està relacionat amb la repetició periòdica (amb petites variacions) de patrons musicals.
- **Melodia**: És una successió de diferents tons musicals que són percebuts com un tot.
- **Harmonia** : És la organització a través del temps de diferents sons amb un pitch similar.
- **Estructura**: És la composició al llarg del temps de totes les dimensions anteriors. Conté repeticions, silencis, canvis de tempo i altres variacions.

Com que el nostre objectiu és identificar peces musicals, i no només sons, anem a veure quines són les característiques que podem extreure d'una peça, i que ens serien útils a l'hora de discriminar-les.

El timbre, l'orquestració i l'acústica són característiques més relacionades amb la percepció del so, i són característiques de curt abast. El seu processament és dut a terme pel sistema auditiu humà tenint una mostra de pocs mili-segons. En moltes peces musicals, aquests canvien ràpidament en el temps. Aquestes característiques les anomenarem de **curt termini.**

El ritme, la melodia i l'harmonia ens informe sobre com estan combinats els sons. Són usualment descrits per conjunts de notes, que són sons únics produïts per un mateix instrument. Aquestes característiques ens informen sobre característiques de la peça musical per tant direm que són de **mig termini**.

Finalment l'estructura, ens informa de com es troben disposades les característiques anteriors al llarg del temps, per tant aquesta és una característica de **llarg termini**.

Un cop una peça ha estat creada, hi ha altres característiques que estableixen altres paràmetres a part dels vistos anteriorment, que especifiquen com ha de ser executada , aquestes són:

- **Tempo** : És la velocitat en la qual una peça musical és interpretada, es sol mesurar en pulsacions per minut, també conegut per les seves inicials en anglès BPM o *Beats per minute*.
- **Tonalitat** : Entenem tonalitat com la freqüència base, sobre la qual es defineixen la resta de freqüències de la peça. És un paràmetre que estableix l'agudesa o gravetat de la peça en general, i pot ser alterada a l'hora d'executar-la.

2.4 Manipulació digital del so

Ja hem vist com la música pot ser classificada i identificada a través de diferents característiques que ens ofereix el llenguatge musical, però, com podem extreure aquestes característiques? En aquest apartat, veurem quins són els instruments bàsics que tenim a l'hora d'obtenir, i manipular sons.

En primer lloc veurem com ho fem per a capturar un so i poder expressar-lo digitalment per a treballar-hi. Posteriorment veurem algunes representacions alternatives a la temporal que ens seran força útils a l'hora d'extreure'n característiques. I finalment veurem quins són els diferents estàndards utilitzats a l'hora d'emmagatzemar aquests sons en arxius d'ús quotidià.

2.4.1 Digitalització

La digitalització és el procés mitjançant el qual, obtenim un senyal digital a partir d'un senyal analògic. Consta principalment de dues etapes.

La primera etapa, el mostreig, tracta d'obtenir valors del so a intervals concrets. Un cop tenim aquests valor, la quantificació, s'encarrega d'assignar-los-hi un valor binari, en funció de la seva intensitat.

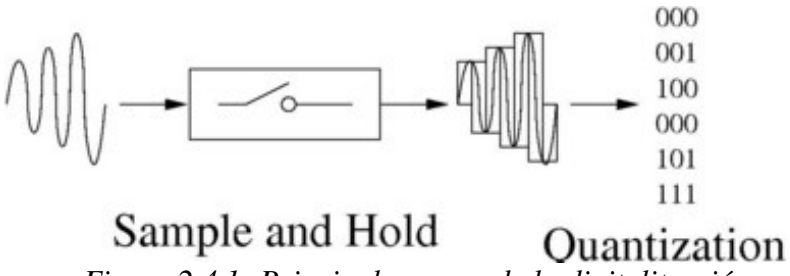


Figura 2.4.1: Principals passos de la digitalització

Veiem ara cadascun d'aquests passos més profundament.

2.4.1.1 Mostreig

El mostreig és el pas, en el qual prenem mostres del so a intervals equidistants. Aquest pas ve especificat per la freqüència de mostreig, que ens diu quin és el número de mostres que capturarem per segon.

Aquesta dada serà imprescindible per a poder reproduir posteriorment el so, i és coneguda en anglès per *samplerate*.

Aliasing

L'aliasing és un fenomen que es pot donar en el procés de mostreig, i que fa que el senyal original, no pugui ser reconstruït fidedignament a partir del digitalitzat.

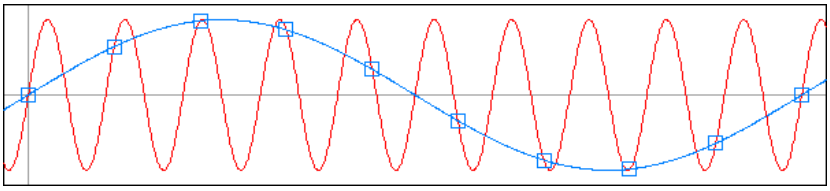


Figura 2.4.2: Aliasing en una ona sinusoïdal

Un exemple ben clar, són les rodes dels cotxes, en moviment. Sabem quin és el sentit en el que roden, i també la seva velocitat, però a vegades ens sembla que girin molt a poc a poc, i en un sentit contrari. Això és degut a l'aliasing, ja que el nostre ull, capta imatges amb una freqüència molt menor a la que seria necessària per a capturar exactament el moviment de la roda, i fa que el que veiem no es correspongui amb la realitat.

A l'hora de mostrejar una mostra de so passa una cosa semblant, si la freqüència de mostreig no és prou elevada, podríem obtenir senyals digitalitzats que ens semblarien molt diferents dels originals.

El teorema de Nyquist, o de Nyquist-Shannon, ens diu que la freqüència de mostreig mínima per a evitar l'aliasing en una mostra de so, ha de ser superior al doble de la màxima freqüència que contingui el senyal que volem digitalitzar.

A la pràctica, com que sabem que la freqüència màxima que podem percebre els humans és d'uns 20 KHz, n'hi hauria prou amb fer servir una freqüència de mostreig de 40 KHz, per a fer indistingible un senyal analògic de la seva digitalització. En efecte, la qualitat de l'estàndard CD-A o *audio compact disc* és 44100 Hz.

2.4.1.2 Quantització

En aquest pas el que fem és assignar un valor binari dins d'un rang prèviament establert al voltatge obtingut del pas anterior.

En aquest pas especifiquem el rang de valors que podrà prendre la intensitat del so en cada mostra, i ho expressem en forma de bits.

Aquesta dada serà imprescindible a l'hora de reproduïr el so, i és coneguda com taxa de bits, o *bitrate*, en anglès.

Els valors usats normalment són 16 bits, o 32 bits, el que fa un rang de $2^{16} = 65536$, o $2^{32} = 4294967296$ valors respectivament.

2.4.2 Formats d'emmagatzemament de sons

En aquest apartat, veurem quins són els formats més utilitats a l'hora d'emmagatzemar documents sonors en un sistema d'arxius.

2.4.2.1 Wav

Wav (WAVE, de l'anglès Waveform Audio Format) és un format creat per Microsoft© i IBM©. Es caracteritza per guardar la música en cru, però afegeix algunes dades addicionals que permeten la seva reproducció.

És força usat en entorns professionals, ja que en no estar comprimit, conserva la qualitat del so.

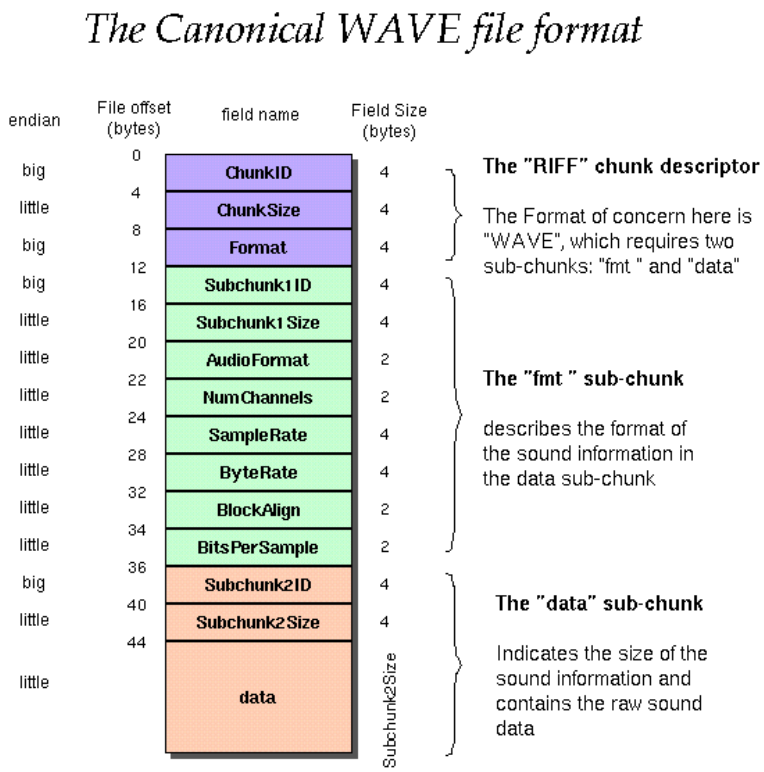


Figura 2.4.3: Camps d'un arxiu WAV

2.4.2.2 mp3

Mp3 o MPEG-1 Audio Layer III és un format d'àudio digital patentat basat en la compressió amb pèrdues i desenvolupat pel grup d'experts de l'MPEG (Moving Picture Experts Group) i descrit per una norma ISO.

Aquest algoritme s'ha convertit en un estàndard de facto, per la seva bona relació entre la compressió que aconsegueixi la pèrdua audible que sofreix.

L'objectiu de l'algoritme de compressió d'aquest format és poder comprimir qualsevol senyal que estigui destinat a ser escoltat i explotar al màxim les limitacions del sistema auditiu humà.

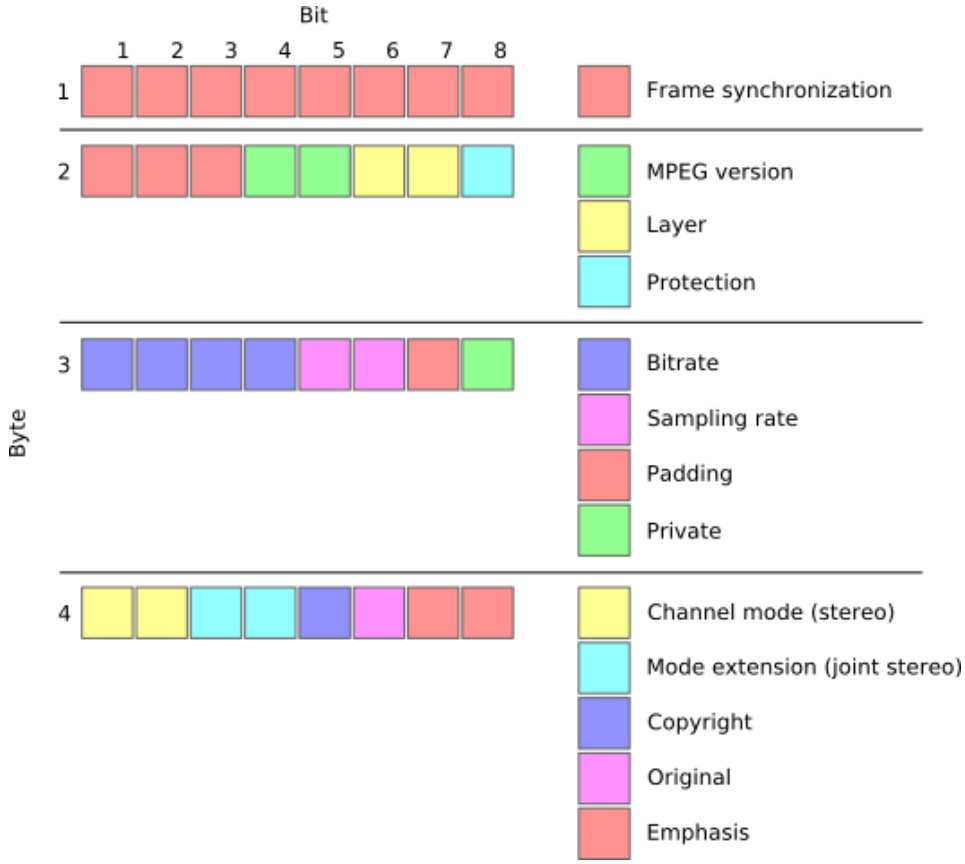


Figura 2.4.4: Format de la capçalera d'un arxiu mp3

2.5 Representació Freqüencial

Fins ara hem vist com la representació d'un so consistia en una sèrie de valors, que indicaven l'amplada de l'ona de so en un instant concret. Aquesta és la representació que anomenem temporal, és a dir, al llarg del temps.

Una representació alternativa a la del domini temporal per als senyals de so, és la representació en el domini freqüencial.

Aquesta representació és molt útil, ja que ens mostra la freqüència fonamental del so (la més petita de totes les freqüències que el formen) i totes les altres que formen el so. És força útil, ja que permet tenir una informació que en la representació temporal no és fàcil de veure.

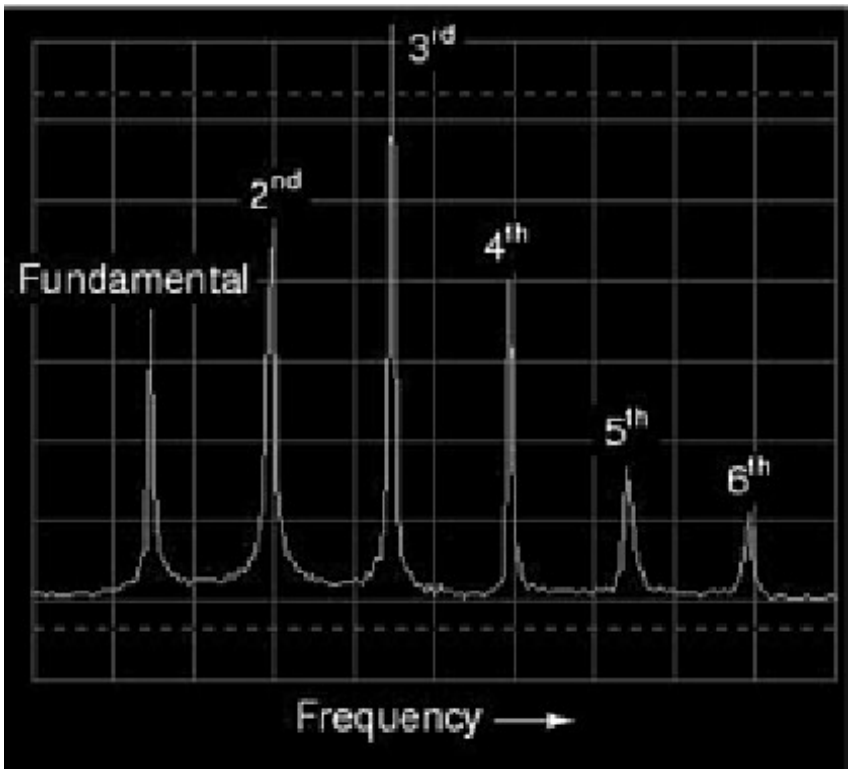


Figura 2.5.1: Espectre d'un so pur (format només per la freqüència fonamental i harmònics d'aquesta)

Sovint també és útil veure una representació de l'evolució de l'espectre al llarg del temps. Però llavors tenim tres dimensions per a representar, a part de cada freqüència i la seva amplada també hem de representar el temps. Això ho podem resoldre mostrant un gràfic tridimensional, o també assignant un color a a cada amplada, en tal cas l'eix X representa el temps, l'eix Y la freqüència, i el color l'amplada de la freqüència a cada instant. A aquesta representació se la sol anomenar espectrograma, o sonograma.

Notem que en l'eix X calcularem un espectre per cada conjunt de n mostres (també anomenat mida de la finestra), haurem de tenir en compte aquest valor a l'hora de triar quina és la freqüència màxima que es troba representada a l'espectre.

Aquesta representació alternativa és molt útil a l'hora d'extreure característiques dels sons, ja que a part d'oferir un punt de vista molt diferent sobre el senyal, comprimeix força els components més rellevants d'aquest.

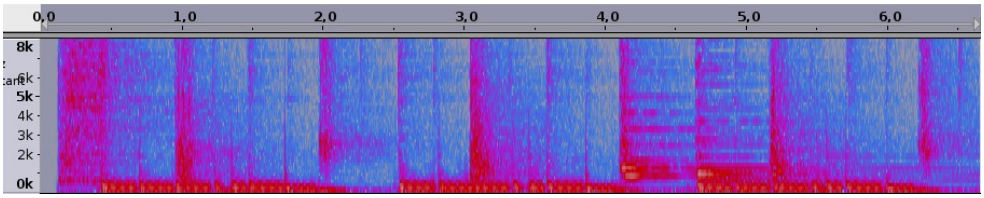


Figura 2.5.2: Espectrograma d'un so

2.5.1 Espectre de potència (o de magnitud)

Anomenem espectre de potència, o de magnitud, a la representació gràfica de les freqüències contingudes en una mostra de so.

2.5.2 Espectre d'harmònics

És com l'espectre de potència, però només les freqüències dels harmònics (és a dir, múltiples de la freqüència fonamental) hi són representats.

2.5.3 Espectre de pics

És com l'espectre de potència, però només les freqüències que tenen amplades majors que cert llindar hi són representades.

Aquest llindar s'expressa amb un valor entre 0 i 1, i es calcula respecte a l'amplada de la freqüència fonamental.

2.5.4 Transformada de Fourier Discreta

La Transformada de Fourier Discreta (TFD), és l'eina matemàtica que ens permet passar de la representació temporal d'un senyal a la freqüencial. Té la característica, que en comptes de representar el senyal com una seqüència de valors, la representa en forma d'una suma de funcions exponencials. Aquesta és l'eina que ens permet calcular l'espectre d'un so.

La fórmula que fem servir per a calcular-la és la següent,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

Figura 2.5.3: Transformada de Fourier Discreta

On x_n representa els valors que pren el vector so al llarg del temps, e és el numero e (2,7182...) i N és la mida del vector original.

Finalment obtindrem una seqüència de N elements, que anomenarem coeficients de Fourier.

La seva inversa és :

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \qquad n = 0, \ldots, N-1.$$

Figura 2.5.4: Transformada de Fourier Discreta Inversa

2.5.5 Transformada de Fourier discreta ràpida (FFT)

És un mètode de càlcul de la TFD, que aprofita certs paral·lelismes en el càlcul de TFDs de seqüències amb un numero d'elements potència de dos. De manera que es reutilitza una gran part dels càlculs, aconseguint reduir la complexitat d'$O(N^2)$ a $O(N\cdot \log_2(N))$.

Avui en dia sempre es fa servir aquest mètode, en el cas que el número d'elements dels quals en vulguem calcular la TFD no sigui potència de dos, ho solucionem afegint zeros, fins a aconseguir un número de potència de dos.

2.5.6 Transformada del Cosinus Discreta

La transformada del cosinus discreta, o DCT (de l'anglès, *Discrete Cosinus Transform*) és una transformada similar a la TFD, però amb la particularitat que només consta de funcions cosinus, per la qual cosa no necessitem nombre exponencials complexos (sovint expressats en forma de sinus i cosinus).

Una de les seves principals propietats i on rau la seva gran força, és que realitza la compactació d'energia força millor que no pas la TFD, per la qual cosa descriu millor un senyal amb el mateix número de components. .

Això fa que sigui molt usada, sobretot en tècniques de compressió, com per exemple en la compressió d'imatges JPEG.

Existeixen diverses variants de DCT, però les més usades són la DCT-II, i la inversa d'aquesta, la DCT-III:

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1.$$

Figura 2.5.5: Transformada del Cosinus Discreta (DCT-II)

Aquesta funció té per entrada els elements del vector de so, i per sortida els coeficients de la transformada.

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n \left(k + \frac{1}{2}\right)\right] \qquad k = 0, \ldots, N-1.$$

Figura 2.5.6: Transformada del Cosinus Inversa Discreta (DCT-III)

2.6 Filtres Digitals

En aquest apartat veurem què són els filtres digitals, quins tipus ens en podem trobar, com funcionen, i finalment, perquè son tan utilitzats en tots els camps del processament del senyal.

2.6.1 Introducció als filtres digitals

Anomenarem filtre digital als sistemes que mitjançant operacions matemàtiques, aconsegueixen alterar les característiques d'un senyal.

Els filtres són caracteritzats exclusivament per la seva funció de transferència, que ens informa de com reacciona un filtre a una entrada.

Alternativament un filtre també pot ser representat per la seva resposta impulsional, que ens permet veure el seu efecte en el domini de la freqüència. Aquesta representació es pot obtenir calculant la transformada de Fourier de la funció de transferència.

Una altra representació que ens pot servir per a tenir una idea del funcionament del filtre és la gràfica, que mostra en l'eix X un rang de freqüències, i en l'eix Y, que ens informa si la banda sofreix algun tipus d'atenuació.

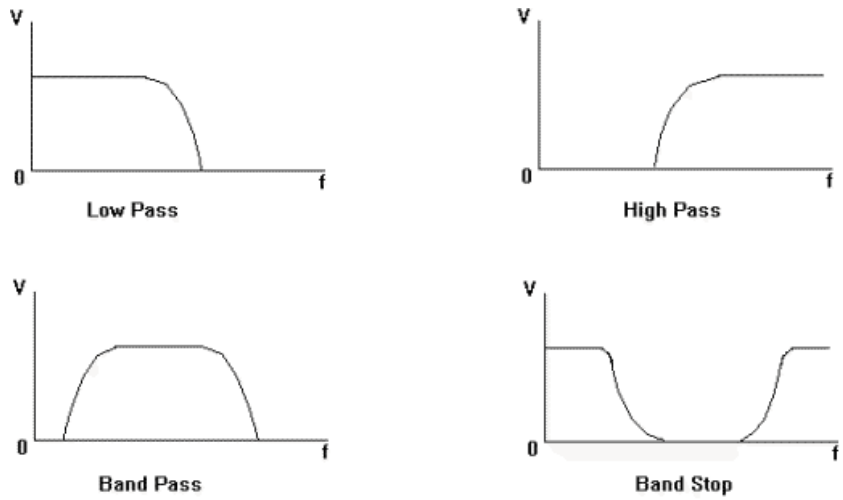


Figura 2.6.1: Corbes de respostes de filtres passa-baixos, passa-alts, passa banda i aturador de banda

La gran potència dels filtres digitals, rau en la capacitat de realitzar operacions que en el domini temporal són extremadament costoses, en el camp de la freqüència, on són sovint més assequibles.

Per exemple, una operació força recurrent dels filtres, la convolució de dos senyals de mida N, la complexitat requerida en el domini temporal és de $O(N^2)$, ja que cal multiplicar cada element d'un senyal per cada element de l'altre. Aquesta operació, en canvi, en el domini freqüencial, té un cost de només O(N).

2.6.2 Aplicacions dels filtres digitals

Una de les aplicacions més recurrents dels filtres és la selecció d'un rang de l'espectre de freqüències, d'aquesta manera existeixen filtres passa alts, que eliminen les freqüències més baixes, els passa baixos, que eliminen les més altes, i els passa banda, que només deixen passar un rang de freqüències determinat.

Els filtres digitals ens poden ser molt útils a l'hora de netejar un senyal, per exemple per a eliminar freqüències que no ens interessin degut al poc impacte que tenen en el sistema auditiu humà, o també per ressaltar algunes característiques.

3 Estat de l'art

3.1 Introducció

L'evolució en els últims decennis de les tecnologies relacionades amb el processament de la informació han suposat una revolució sense precedents en l'automatització de diferents processos.

Entre aquests processos, podem trobar-hi els relacionats amb els mitjans audiovisuals, que pel seu gran impacte en el món actual, han rebut una atenció considerable.

Una de les noves disciplines creades, ha estat la relacionada amb la música, anomenada *Music Information Retrieval.* Aquesta disciplina tracta de resoldre problemes, com per exemple, classificar una llibreria en funció de l'estil de música o proposar nous grups que ens poden agradar en funció de les nostres preferències.

Un dels camps que ha rebut més atenció per part d'aquesta nova disciplina, és el que anomenem recuperació de música basant-se en el contingut, o "*Content-based music retrieval*" en anglès. Aquest camp tracta de calcular la similitud entre diferents sons, amb la finalitat, principalment. de poder posteriorment recuperar informació addicional en una base de dades.

La part principal d'aquest camp, tracta de trobar quin és el millor sistema per a compactar la informació més rellevant d'una mostra de so, de manera que sigui factible guardar-ho en una base de dades, per a posteriorment, realitzar una cerca. Aquesta tècnica és coneguda amb el nom d'àudio fingerprinting.

3.2 Àudio fingerprinting

Anomenem *audio fingerprinting* al resultat del procés mitjançant el qual convertim una mostra de so, en una signatura o conjunt de bits de manera que posteriorment aquesta signatura conservi informació rellevant sobre el so original, amb la finalitat última de que ens serveixi per a discriminar-lo de la resta de sons.

És important també, que les característiques d'aquest fingerprint siguin immunes al soroll, de manera que identifiqui correctament una cançó, independentment de quina hagi estat la seva font de captura: un estudi professional, un telèfon mòbil...

En aquest procés caldrà veure quines són les característiques més rellevants a l'hora d'identificar inequívocament uns mostra de so, així com el cost computacional del seu càlcul.

3.3 Propietats del fingerprint

Les propietats dependran en gran mesura dels requeriments de l'aplicació final, ja que per exemple, no serà el mateix fer una aplicació per a reconèixer ordres de veu, que per a identificar el catàleg d'una companyia de gestió de drets d'autor. Veiem a continuació, quines són les propietats desitjables que podria tenir una aplicació final [9]:

- **Precisió** : La taxa entre encerts i falsos positius. No és el mateix una aplicació per a omplir les metadades d'una biblioteca musical, que una per a comprovar que una cançó no pertany al catàleg d'una companyia de gestió de drets d'autor.

- **Robustesa** : L'encert del sistema en funció de soroll ambient. No és el mateix una aplicació per a omplir les metadades d'una biblioteca musical, on els arxius no tenen cap mena de distorsió, que una aplicació per descobrir el nom d'una cançó que estem sentint a través d'un servei que funcioni a través d'un telèfon mòbil.

- **Granularitat**: Quina durada de temps tenim per a identificar la cançó? En aquest cas els exemples de la propietat anterior també serveixen. En l'aplicació per a omplir metadades tenim la cançó completa, mentre que en l'altra tindrem uns pocs segons.
- **Seguretat**: Vulnerabilitat de l'aplicació a la manipulació. Els exemples anteriors també serveixen. Si volem controlar els nostres drets d'autor, probablement haurem d'anar amb compte amb versions de les cançons específicament manipulades per a no ser reconegudes pel nostre sistema.
- **Versatilitat**: Habilitat d'identificar el so independentment del seu format. El programa per omplir les metadades, seria útil que acceptés diferents formats. En canvi, un programa que tingui com a única entrada d'àudio un micròfon o un telèfon, sempre tindrà el mateix format d'entrada.
- **Escalabilitat**: Rendiment de l'aplicació amb un gran numero d'elements a la base de dades o amb una gran quantitat de transaccions concurrents. El sistema de recuperació a través del telèfon mòbil no serà el mateix, per exemple, que un sistema domèstic de reconeixement d'ordres de veu.
- **Complexitat**: Quin és el cost de crear un fingerprint, o d'afegir un fingerprint a la base de dades? Això va en funció de l'arquitectura interna de cada aplicació.

Pel que fa a la nostra aplicació, l'objectiu principal serà analitzar quin dels algoritmes ens ofereix una millor **precisió**, és a dir, quin identifica correctament un major nombre de cançons, per tant aquesta serà una propietat a tenir en compte.

Alhora, volem que sigui força immune al soroll, per tant també ens interessarà que sigui **robusta**.

Respecte a la **granularitat,** doncs no ens preocupa massa, ja que com veurem posteriorment, treballarem amb segments d'una durada reduïda.

La **seguretat** no és pas una prioritat, ja que en tractar-se d'una comparativa, treballarem en entorns tancats.

La **versatilitat** tampoc ens preocupa per la mateixa raó que l'anterior, treballarem en un entorn tancat.

L'**escalabilitat** tampoc és una prioritat, ja que no treballarem amb una gran quantitat de mostres.

També tindrem en compte la **complexitat** de cadascun dels algoritmes que implementem,

però no serà una prioritat en el desenvolupament.

Un cop tenim clar com ha de ser la nostra aplicació, veiem quines són les propietats desitjables del fingerprint [2]:

- **Resum Perceptual**: Ha de conservar les característiques més rellevants del so.
- **Robust**: Ha de ser immune a sorolls.
- **Compacte**: Cal que sigui de mida reduïda, ja que la mida tindrà un impacte directe en el temps de cerca a la base de dades.
- **De baixa complexitat**: Convé que la seva generació no sigui gaire costosa computacionalment.

També cal tenir en compte, que la semblança que percebem els humans, no és la mateixa que pot percebre una màquina, dit d'una altra manera, el que per a una màquina pot ser una gran diferència, un humà podria ni adonar-se'n. Això és degut a que el sistema auditiu humà té una estructura, que fa que la seva percepció sigui única.

Per a intentar estudiar aquesta percepció diversos investigadors han creat models informàtics que pretenen simular el funcionament del sistema auditiu humà. És a dir, conjunts de funcions, que intenten modelar el funcionament dels òrgans interns, de manera que el que arriba finalment al nostre cervell mitjançant els nervis, sigui el més semblant a la sortida d'aquests sistemes.

Una de les característiques més rellevants relacionades amb la diferent percepció dels sons, és la diferent sensibilitat a les diferents freqüències que tenim els humans degut al funcionament del nostre sistema auditiu. Tenir això en compte sol ser una bona idea de cara a millorar les taxes d'encert de diferents algoritmes.

3.3.1 Escenaris d'aplicació

Els escenaris d'aplicació d'un sistema capaç d'identificar una cançó a partir simplement d'escoltar-la són molts, a continuació en mostrarem alguns dels més comuns.

Validació de metadades d'una biblioteca: Sovint molts arxius musicals tenen metadades errònies, o directament no en tenen. El nostre sistema podria servir per a omplir correctament el camp de metadades de diversos arxius, és a dir, per exemple en el cas de les cançons l'artista, el títol de la cançó, l'estil musical al qual pertany...

Validació de repertori: L'àudio fingerprinting, pot ajudar a una entitat de gestió de drets d'autor, a comprovar que una emissora de ràdio posseïx totes les llicències de les cançons que emet.

Identificació de peces en un concert: Volem per exemple saber si un grup en un concert van tocar certa cançó. L'àudio fingerprinting ens permetria obtenir una resposta automàticament.

Serveis de valor afegit: Per exemple, podria existir un servei per a guitarristes professionals, que mostrés els acords d'una cançó, un cop aquesta ha estat correctament identificada. O per a una emissora de ràdio, un sistema que digui si una cançó té drets d'autor, i en cas afirmatiu qui n'és el dipositari. També oferir les lletres d'una cançó, per tal de poder cantar-la com si fos un karaoke.

3.4 Estat Actual de l'àudio fingerprinting

Actualment hi ha diversos sistemes que resolen el problema. A continuació veiem alguns detalls dels més populars.

3.4.1 Shazam <http://www.shazam.com>

Es basa en diferents característiques que veurem posteriorment, fa servir la mitjana, variància, obliqüitat, curtosis, Jarque-Bera test (valor calculat en funció de la obliqüitat i la curtosi), i la correlació dels 20 primers components del test Portmanteau de Ljung-Box.

3.4.2 A highly robust audio fingerprintg System [3]

Fa servir els primers coeficients de Fourier, els coeficients cepstrals de les freqüències de Mel, l'amplada espectral, l'agudesa de l'espectre, el codificador de predicció lineal així com característiques més comunes, com la intensitat del senyal.

3.4.3 TRM Recognizes Music <http://www.relatable.com/tech/trm.html>

Fa servir la mitjana de creuaments amb zero, pulsacions estimades per minut, mitjana de l'espectre i altres característiques més comunes. Tots ells són computacionalment eficients, i produeixen fingerprints de mida força reduïda.

3.4.4 Features for audio and music classification [10]

L'alt cost de processament de totes aquestes característiques ha fet que s'hagin buscat alternatives, que en alguns casos permeten obtenir millors resultats.

Per exemple [4] tracta d'una representació de l'envolvent temporal que intenta representar el sistema auditiu humà i funciona de la següent manera: Cada mostra es filtra mitjançant un conjunt de filtres *gammatone* (filtres lineals descrits per una resposta impulsional que resulta del producte d'una distribució gamma i d'un to sinusoïdal), que representa la resolució

espectral del sistema auditiu perifèric. Posteriorment un anàlisi temporal és calculat mitjançant la modulació de l'espectre de l'envolvent del resultat de l'aplicació de cada filtre.

Un cop calculat l'espectre de la modulació de cada filtre aquest és normalitzat pel valor mitjà (freqüència de 0 Hz) i posteriorment parametritzat per la suma de les energies en quatre bandes de freqüències i fent el logaritme. Aquestes quatre bandes representen les freqüències següents: 0 Hz, 3-15 Hz , 20-150 Hz i 150-1000 Hz.

Finalment agafem com a característiques l'energia que modula l'envolvent de cadascuna d'aquestes bandes.

3.4.5 PRH (Philips Robust Hash) [7]

Últimament s'han ideat noves alternatives als sistemes basats en característiques, principalment degut, a l'alt cost computacional que requereix la seva extracció. Els resultats d'aquestes noves alternatives semblen ser molt superiors als obtinguts mitjançant l'extracció de característiques habituals fins al moment. Un d'ells és el Philips Robush Hash, anem a veure com funciona.

Aquest algoritme en primer lloc separa la mostra de so en vàries submostres solapades. Un cop fet això es calcula l'espectre de potència de cadascuna d'elles, i es separa en diferents bandes de freqüència logarítimicament espaiades, per a tenir en compte el funcionament del sistema auditiu humà. A partir d'aquí, es calcula l'energia total de cadascuna de les subbandes al llarg de les diferents submostres, i amb aquestes dades es calcula el fingerprint final.

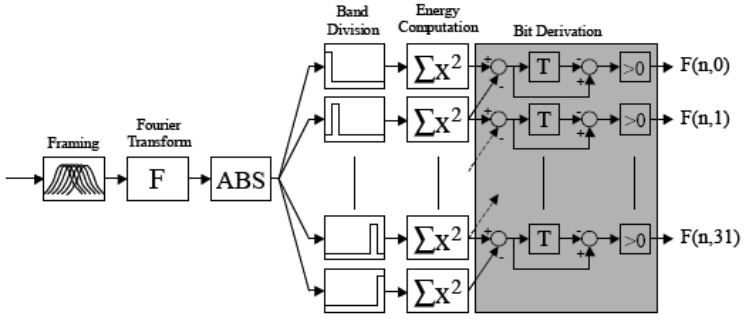


Figura 3.4.1: Descripció gràfica del l'algoritme PRH

Tot i la simplicitat d'aquesta idea, la taxa de recuperació de fingerprints és força elevada i el seu càlcul força senzill computacionalment.

3.4.6 Basat en la DCT [11]

Aquest algoritme és força semblant al PRH, però va una mica més enllà.

Mentre que el PRH calcula el fingerprint de les submostres directament a partir de l'energia de cadascuna de les bandes, el basat en la DCT calcula la DCT de la sèrie d'aquestes energies, i és amb aquestes noves dades que genera el fingerprint.

Amb això es redueix la variació del fingerprint per efecte del soroll, i la seva taxa d'encerts resulta bastant superior a la del PRH.

3.5 Passos comuns als diferents algoritmes d'àudio fingerprinting

Tot i les grans diferències existents entre els diferents algoritmes que hem citat, podem identificar una sèrie de passos comuns a la majoria de tots ells, que ens permetran fer-nos una idea del seu funcionament.

En primer lloc el es fa és el **preprocessament**, es remostreja el so a una freqüència determinada, i s'apliquen alguns filtres per a eliminar els sorolls més comuns, intentant perdre la mínima quantitat d'informació rellevant, i si el so d'entrada es troba en estèreo, es converteix a mono.

Posteriorment el que fem és la **segmentació** del so, és a dir, dividim la cançó en petites parts, de manera que aquestes parts siguin fixes, i siguin les mateixes tant en la cançó completa com en una mostra de la cançó. Això ens permetrà no haver de conèixer la cançó sencera per a identificar-ne una part.

Un cop tenim la mostra d'àudio segmentada, en calculem les diferents **transformades** (DFT, DCT...), que ens serviran per al següent pas, **l'extracció de característiques**.

Posteriorment ens trobem amb el punt més rellevant, el **càlcul del fingerprint**. Aquest pas extreu les característiques més rellevants i discriminatòries d'un segment per tal de comprimir la informació que conté en la menor mida possible.

Finalment la **indexació**, organitza tots els fingerprints de la base de dades, per tal de que la cerca sigui el més ràpida possible.

Un cop haguem trobat quin és el segment de la base de dades més similar, aquest haurà d'estar associat a un conjunt de metadades, (e.g: Artista, Títol, Gènere...) que seran la resposta del sistema.

Veiem ara, cadascuna d'aquestes etapes amb una mica més de detall.

3.5.1 Segmentació

La segmentació és el procés que ens permet partir una mostra de so en diferents fragments per tal de poder d'aquesta manera recuperar la cançó sense haver de processar-la completament. És important que aquesta separació estigui basada en les característiques del so, ja que si ens donen una mostra de so, la segmentació haurà de ser igual en la mostra que en la cançó original.

És important tenir en compte la durada desitjada dels segments, ja que això tindrà un impacte elevat sobre la funcionalitat del sistema. Si triem segments petits, la quantitat d'informació que ens proporcionin serà minsa, però si els triem molt grans, llavors

- Necessitarem mostres d'àudio de major durada
- Serà més ràpid trobar fingerprints a la base de dades.

- Tindrem una taxa d'encerts més elevada (el fingerprint contindrà més informació)
- El càlcul del fingerprint serà més costós.

Actualment els sistemes treballen amb segments d'entre 20 i 200 ms, a primera vista, aquesta durada és excessivament minsa, ja que l'objectiu del sistema és tenir alguns segons per a identificar una cançó, a la pràctica, s'ha vist que agafar segments de durades superiors implica una complexitat computacional inassolible.

A la pràctica el que es sol fer és decidir una característica o conjunt d'elles, d'alguna manera que els pics es trobin equiespaiats, un cop fet això, els pics seran les separacions entre segments, o el centre dels segments.

Per exemple en la música rock, és comú que el ritme sigui marcat amb el tambor principal de la bateria, que té un impacte freqüencial força característic. Una opció és identificar aquests canvis al llarg del so, calculant l'espectre de petites parts del so (com ho fem per a crear un espectrograma), i fer-los servir com a separadors de segments.

3.5.2 Preprocessament

Un cop ens arriba una mostra d'àudio el que es fa és descomprimir-la (si es troba comprimida), passar-la a mono (si està en estèreo) i finalment remostrejar-la a una freqüència que ens permeti reduir la seva mida, però conservant la informació que ens ha de permetre identificar-la. La freqüència de remostreig sol moure's entre els 5 i els 11KHz.

Aquest pas és comú en tots els sistemes estudiats.

Alguns sistemes adicionalment apliquen algun filtre per a eliminar freqüències que hagin pogut resistir al submostreig. Alguns filtres també poden servir per a eliminar sorolls comuns.

3.5.3 Transformades

En aquest pas el que fem és calcular diverses transformades o representacions alternatives del senyal expressat en el domini temporal.
Existeixen transformades òptimes, en el sentit de compressió de la informació i de la seva correlació, però sovint el seu càlcul té un cost computacional que les fa inassumibles.
Per aquesta raó, altres transformades el càlcul de les quals es fa mitjançant vectors comuns són usades, com per exemple la DFT o la DCT.

3.5.4 Extracció de característiques

En aquest pas extraiem la informació més rellevant del segment, ja sigui a partir de la seva representació temporal, o d'alguna altra de les transformades prèviament calculades.

Com que el que busquem són característiques que mesurin la semblança a nivell perceptiu, són comuns sistemes que extreguin característiques fent un previ anàlisi de diferents bandes de l'espectre, per exemple els Coeficients Cepstrals de les freqüències de Mel, o MFCC per les seves inicials en anglès (*Mel frequency Cepstral Coefficents*).

Existeixen moltes de característiques que podem usar, i contínuament diversos investigadors en proposen de noves, però convé que compleixin les següents propietats [7]:

- Càlcul poc complex
- Alta variància
- Poca correlació amb altres característiques
- Immunes al soroll

3.5.5 Creació del fingerprint

Un cop ja tenim el valor de totes les característiques, és el moment de crear el fingerprint. El fingerprint, serà l'identificador del segment, que guardarem, i que actuarà com a clau de cerca. Per tant és interessant que mantingui tota la informació possible de les característiques, així com que la seva mida sigui el més petita possible.

El tipus de fingerprint més elemental és en forma de vector multidimensional, on cada dimensió representa una característica.

Aquesta alternativa però és força redundant, ja que carrega el fingerprint amb informació que pot no ser rellevant a l'hora de discriminar entre fingerprints. Per això es proposen una sèrie de mètodes per a intentar conservar només la informació més rellevant de cara a discriminar els fingerprints.

PCA

Una de les tècniques més usades per a extreure només la informació més rellevant a l'hora de discriminar fingerprints és la reducció de la dimensionalitat, i un dels mètodes més comuns és l'anàlisi de components principals, anomenat PCA (*Principal Components Analyisis*) per les seves inicials en anglès.

Aquesta tècnica construeix una transformació lineal, que escull un nou sistema de coordenades per al conjunt original de dades, de manera que la major variància del conjunt de dades és capturada en el primer eix (anomenat primer component principal), la segona serà el segon component, i així successivament.

Això a la pràctica el que aconsegueix, és codificar no els valors de les característiques, sinó les diferències que hi ha entre elles. D'aquesta manera, si per exemple abans teníem un vector amb 10 característiques, ara podem representar el valor d'aquestes característiques amb 7 elements, i aconseguir mantenir la mateixa informació. Una altra manera d'entendre el que fa el PCA és pensar que elimina tota la informació que és comuna a les característiques dels

diferents segments, és a dir, elimina tota correlació.

Aquest sistema però, és força costós computacionalment, ja que ha de tenir en compte els fingerprints de tots els segments, si bé això és costós, també és assumible. El que no és tant assumible, és que per a garantir la millor efectivitat de l'algoritme, aquest ha de ser recalculat cada vegada que s'afegeix un fingerprint a la base de dades, la qual cosa té un impacte molt elevat en l'escalabilitat del conjunt de fingerprints.

Codificació de l'evolució de les característiques

Una altra alternativa per a reduir la mida del fingerprint és codificar no el valor de cada característica, sinó valors que modelin el seu comportament al llarg del segment. Per exemple, la primera derivada de la funció que modela la seva evolució, o els primers components de la transformada de Fourier, o de la transformada del cosinus.

Això té alguns avantatges força interessants, per exemple que és relativament immune al soroll si aquest és constant, i que augmenta significativament la taxa d'èxit, ja que la informació que finalment es compara és molt superior, i no afecta a la mida del fingerprint.

Aquest sistema però requereix una capacitat de càlcul molt superior, ja que multiplica el càlcul de cadascuna de les característiques per tants subsegments com fem, i posteriorment necessita calcular la transformada per a cada característica.

Basat en diccionari

Altres tècniques, relacionades amb algoritmes usats en el reconeixement de la parla, codifiquen en primer lloc l'evolució de diferents característiques el llarg d'un segment, i intenten trobar-hi patrons comuns, i aquestes són les dades que s'usen per a fer el fingerprint. Fent una similitud amb la parla, podem dir que hi ha unes lletres (variació concreta d'una característica) i que totes les cançons estan compostes per diferents lletres. Aquest sistema però s'ha demostrat força costós, ja que el conjunt de lletres que cal per a ser efectiu és força elevat.

3.5.6 Indexació

Un cop tenim el fingerprint de cadascun dels segments, és el moment de guardar-lo al catàleg, i a partir d'aquí ja podríem començar buscar coincidències.

La gran quantitat de fingerprints que podem arribar a tenir a la base de dades, fa que la comparació seqüencial de fingerprints , no sigui una opció i per tant s'hagin de trobar sistemes d'indexació, per mica en mica, anar apropant-nos al fingerprint més similar.

Veiem a continuació, alguns dels mètodes més usats.

3.5.6.1 Arbres

La tècnica més usada per a la indexació és la basada en arbres, la qual, es basa en la creació d'un arbre binari per a cada característica del so. És a dir, per a cada característica es fan divisions successives en funció del valor que pren, i d'aquesta manera anem discriminant possibles fingerprints. Un cop arribem a un node fulla tindrem un conjunt de possibles fingerprints. Repetim aquest pas per a cada característica, i finalment ens quedem amb els fingerprints que han aparegut a un major nombre de fulles, o a totes, idealment.

Idealment aquest procés, només hauria d'oferir un fingerprint com a resultat, però si n'hi hagués més d'un podríem calcular la distància euclídea de cadascun amb la del fingerprint d'origen, en aquest cas, ja tindrem un numero de fingerprints candidats força baix i el càlcul d'aquesta similitud seria assumible. Per a que funcioni bé aquest sistema, és important que a cada node no-fulla, la funció de decisió discrimini el mateix número de fingerprints per a una opció que per a l'altra, és a dir, que estigui ben balancejat. Altrament el sistema no seria gaire eficient.

3.5.6.2 Hash

L'altre opció que s'ha demostrat força vàlida, és la de les taules de hash. Com que no

podem fer servir el fingerprint directament com a clau de la taula, ja que ens interessa una mesura de similitud, i no el fingerprint exacte, el que es fa és crear tantes taules de hash com coeficients tinguem.

Un cop fet això, a cada entrada de cada taula de hash se li assigna un rang, que està solapat amb els seus veïns, i a partir d'aquí fem la cerca.

Aquesta ens retornarà un numero de candidats per a cada coeficient. Llavors seleccionem el candidat que sigui comú a totes les taules de hash, i ja tenim el fingerprint.

Amb tot és possible que no aparegui el mateix candidat a totes les taules, en tal cas agafem aquell que aparegui més cops, i si n'hi ha més d'un, calculem la distància euclidiana entre els candidats i el fingerprint objectiu.

3.6 Característiques

Anomenem característiques a totes aquelles mesures d'un so, que ens donen algun tipus d'informació sobre com és i que ens permetin de diferenciar-lo d'altres.

Actualment no hi ha gaire acord entre les millors característiques a fer servir a l'hora de discriminar sons, ja que són força específiques del tipus de música que pretenem recuperar. Així per exemple, si ens interessa la música rock, segurament el ritme seria una bona característica a fer servir, però per a la música clàssica no ens serviria gaire.

Podem dividir les característiques en diferents punts de vista, per exemple entre les que són d'alt i baix nivell. Les d'alt nivell són les que podem percebre els humans (ritme, melodia, pulsacions per minut...) mentre que les de baix nivell són perceptibles només per sistemes informàtics (freqüència fonamental, transformada de Fourier, coeficients cepstrals...).

Les característiques d'alt nivell solen ser més eficients a l'hora de discriminar sons, però com que solen aparèixer en intervals de temps força grans, de l'ordre d'alguns segons, el seu càlcul té un cost computacional més elevat, i sovint inassumible.

Per tant ens centrarem en característiques de baix nivell, ja que tot i no ser òptimes permeten aconseguir bons resultats.

A continuació veurem algunes característiques que usen sistemes implementats actualment, classificades segons el seu origen de dades [13].

Per a totes les fórmules citades a continuació, les variables representen les següents dades:

- x_i, Representa l'amplada del senyal en l'instant i
- $freq_n$ i amp_n, Per a cada n, ens diu l'amplada amp_n de la freqüència $freq_n$
- f0, Representa la freqüència fonamental del senyal.

3.6.1 Extretes a partir de la representació temporal

3.6.1.1 Mitjana

És la mitjana dels valors que pren el vector de so.

$$\bar{x} = \frac{1}{N} \cdot \sum_{1}^{N} x_i$$

3.6.1.2 Variància

Ens informa de la dispersió dels valors que pren el vector de so.

$$\mu = \frac{1}{N-1} \cdot \sum_{1}^{N} (x - \bar{x}_i)^2$$

3.6.1.3 Desviació Mitjana

És una altra mesura de dispersió. Aquesta és anàloga a la desviació estàndard, però aquest cop, en comptes de calcular el quadrat de la distància entre l'element i la mitjana, en calculem el valor absolut.

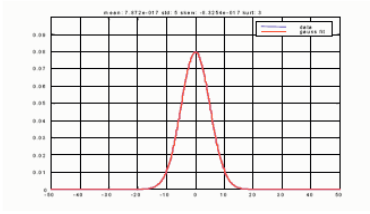
$$D_m = \frac{1}{N} \cdot \sum_{1}^{N} |x - \bar{x}_i|$$

3.6.1.4 Desviació estàndard

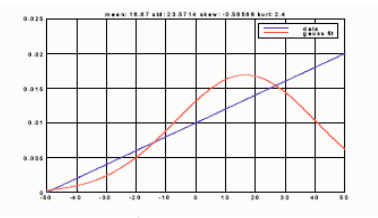
És una altra mesura de dispersió, aquesta totalment relacionada amb la variança.

$$D_e = \sqrt{\mu}$$

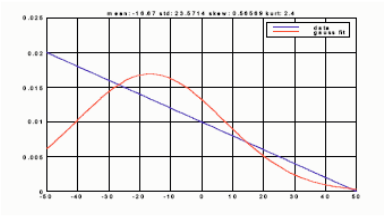
3.6.1.5 Asimetria



skewness=0



skewness<0

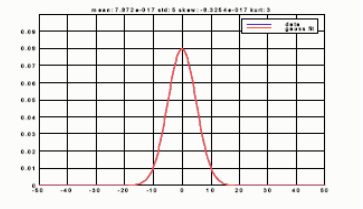


skewness>0

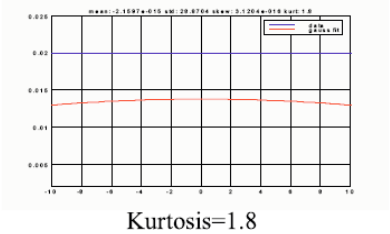
L'asimetria, o *skewness* en anglès, ens informa del grau de simetria que presenta la distribució de valors que pren el vector de so respecte de la seva mitjana. Una asimetria negativa, indica que la majoria de valors tenen valor inferior a la mitjana, és a dir, en la funció de probabilitat de densitat, la majoria de valors es troben a l'esquerra de la mitjana.

$$Asimetria = \frac{1}{N} \cdot \sum_{1}^{N} \left(\frac{x_i - \mu}{\sigma} \right)^3$$

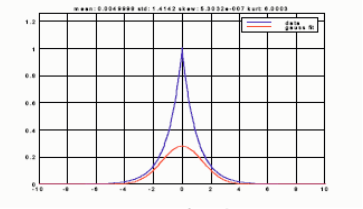
3.6.1.6 Curtosi



Kurtosis=3



Kurtosis=1.8



Kurtosis=6

Ens informa de l'escarpament que prèn la funció de distribució de probabilitat dels valors que pren el vector de so. Com més alt sigui aquest valor, més concentrats es trobaran els valors al voltant de la mitjana.

$$Curtosi = \frac{1}{N} \cdot \sum_{1}^{N} \left(\frac{x_i - \mu}{\sigma} \right)^4$$

3.6.1.7 Mitjana Quadràtica (RMS)

És una mitjana especialment apta per a variables que tenen valors positius i negatius, ja que calcula el quadrat dels valors d'entrada.

Tècnicament és l'arrel quadrada de la de la mitjana dels valors del senyal al quadrat.

$$x_{rms} = \sqrt{\frac{1}{n} \cdot \sum_{1}^{N} x_i^2}$$

3.6.2 Extretes a partir de l'espectre de potència

3.6.2.1 Característiques bàsiques estadístiques

La mitjana, la variància espectral, la desviació estàndard, la desviació mitjana, l'asimetria i la curtosi són característiques que es poden aplicar a les amplades de les freqüències de l'espectre de manera anàloga a com ho fem en la representació temporal del senyal.

3.6.2.2 Centroide

Si prenem l'espectre, com una distribució els valors de la qual són les freqüències i la probabilitat d'aquestes de ser observades són les seves amplades. El centroide és el centre geomètric d'aquesta distribució.

A la pràctica ens permet distingir entre sons percussius, per exemple el so d'un tambor, i sons sostinguts, per exemple un acord de violí.

$$Centroid = \frac{\sum_{1}^{N} freq_n * amp_n}{\sum_{1}^{N} amp_n}$$

3.6.2.3 Irregularitat

Aquesta és una mesura, que ens informa sobre els canvis entre les amplades de l'espectre a curt plaç.

$$Irregularitat = \sum_{k=2}^{N-1} \left| amp_k - \frac{amp_{k-1} + amp_k + amp_{k+1}}{3} \right|$$

3.6.2.4 Suavitat de l'espectre

Ens informa sobre la suavitat de l'espectre. Un espectre amb variacions moderades entre freqüències contigües tindrà suavitat alta, mentre que si es produeixen grans desnivells, aquesta serà baixa.

$$Progpagació = \sum_{2}^{N-1} \left| 20\log amp_n - \frac{20\log amp_{n-1} + 20\log amp_n + 20\log amp_{n+1}}{3} \right|$$

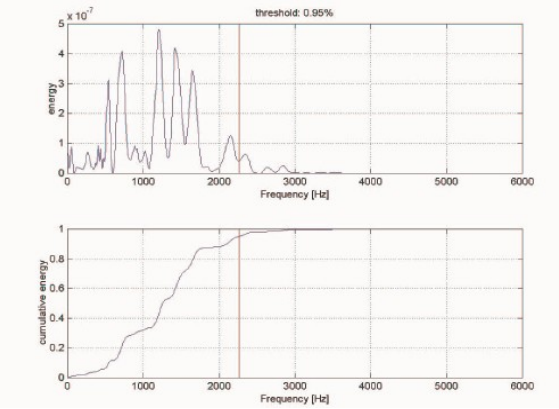
3.6.2.5 Propagació de l'espectre

Ens informa de com es troben repartides les amplades de l'espectre al llarg de les freqüències.

$$Progpagació = \sqrt{\frac{\sum_{1}^{N} (n - centroide)^2 * amp_n}{\sum_{1}^{N} amp_n}}$$

3.6.2.6 Rolloff

Ens informa de com de concentrada es troba l'energia en l'espectre. Concretament, és la freqüència màxima, per sota de la qual hi trobem un cert percentatge del total de l'energia del senyal. Aquest percentatge sol ser el 90%.



$$Total\ Energy = \sum_{1}^{N} amp_n$$

$$Energy\ Thresold = (Total\ Energy) * Thresold$$

$$Rolloff = \frac{Samplerate}{N} * \sum_{1}^{Energy\ Thresold} amp_n$$

3.6.2.7 Planesa de l'espectre (flatness)

Ens informa sobre la tendència a experimentar variacions de l'espectre en freqüències veïnes.

$$Planesa = \frac{[\prod_{1}^{N} amp_n]^{\frac{1}{N}}}{\frac{\sum_{1}^{N} amp_n}{N}}$$

3.6.2.8 Pendent de l'espectre

Ens informa de quina és la pendent de l'espectre

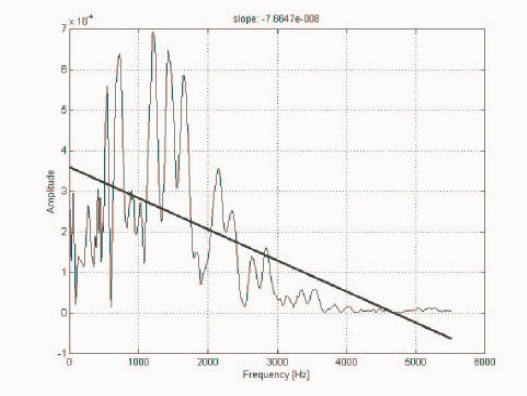
$$Freq_{sum} = \sum_{1}^{N} freq_n;$$

$$Amp_{sum} = \sum_{1}^{N} amp_n;$$

$$FxA_{sum} = \sum_{1}^{N} freq_n * amp_n;$$

$$FreqSQ_{sum} = \sum_{1}^{N} freq_n^2;$$

$$Pendent = \frac{1}{Amp_{sum}} * \frac{N * FxA_{sum} - Freq_{sum} * Amp_{sum}}{N * FreqSQ_{sum} - Freq_{sum} * Freq_{sum}}$$



3.6.2.9 MCFF

També anomenat Coeficients cepstrals de les freqüències de Mel (*Mel-Frequency Cepstral Coefficients)*, és una representació compacta de l'espectre, que té en compte la percepció no linear de les diferents freqüències que fa el sistema auditiu humà.

El que fa és assignar un pes concret a cadascuna de les freqüències dins de cadascuna de les bandes definides per l'escala de Mel.

3.6.3 Extretes a partir de l'espectre de pics

3.6.3.1 Inharmonia Espectral

Ens informa sobre la relació que hi ha entre la freqüència fonamental d'un so i els seus harmònics.

$$Inharmonia = \frac{\sum_{1}^{N} |freq_n - n + f0| * \sqrt{Amp_n}}{\sum_{1}^{N} \sqrt{Amp_n}}$$

3.6.4 Extretes a partir de l'espectre d'harmònics

3.6.4.1 Rati entre senars i parells

Ens informa del rati entre harmònics senars i parells que té el so.

$$Rati\ entre\ senars\ i\ parells = \frac{\wedge(Frequencies\ Senars\ amb\ amplitud\ diferent\ a\ zero)}{\wedge(Frequencies\ Parells\ amb\ amplitud\ diferent\ a\ zero)}$$

3.6.4.2 Triestímuls

Són una mena de port del sistema de colors primaris (RGB) al mon del l'àudio. Pretenen concentrar en tres valors, el pes relatiu dels diferents harmònics d'un so.

El primer triestímul, mesura el pes del primer harmònic, el segon mesura el pes del segon, tercer i quart harmònics, i finalment el tercer triestímul mesura el pes de la resta d'harmònics.

En aquest cas els termes a_i representen l'amplada del l'i-èssim harmònic, i H el numero total d'harmònics que tenim.

$$T1 = \frac{a_1}{\sum_{h=1}^{H} a_h}$$
$$T2 = \frac{a_2 + a_3 + a_4}{\sum_{h=1}^{H} a_h}$$
$$T3 = \frac{\sum_{h=5}^{H} a_h}{\sum_{h=1}^{H} a_h}$$

3.6.5 Relacionades amb les bandes de Bark

A la pràctica si sentim dos sons amb la mateixa amplada, però amb diferents freqüències, el volum que percebem, és diferent.

Les bandes de Bark són una subdivisió del camp freqüencial que relaciona cada freqüència amb la percepció del volum que tenim els humans, de manera que fa una estimació força acurada del volum que percebem.

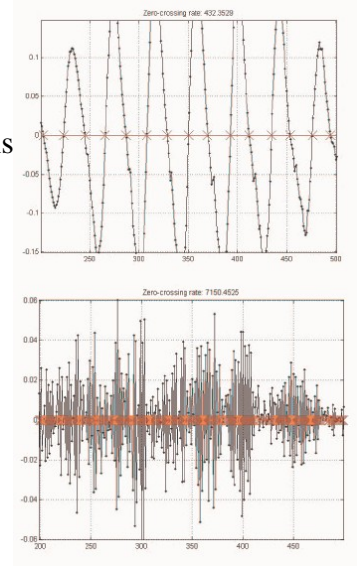
3.6.5.1 Intensitat (Loudness)

Ens informa de la percepció del volum segons el sistema auditiu humà.

$$Intensitat = \sum_{1}^{N} (BarkBand_n)^{0.23}$$

3.6.6 Extretes a partir d'altres característiques

3.6.6.1 Rati de creuaments amb zero



Ens diu quantes vegades el senyal creua l'eix horitzontal. Ens informa de la sorollositat del so.

$$Rati\ de\ creuaments\ amb\ zero = \wedge((x_n * x_{n-1}) < 0)\, x \in (2, N)$$

3.7 Llibreries Candidates

En aquest apartat veurem algunes llibreries que faciliten l'extracció d'algunes de les característiques que hem vist anteriorment. L'ús d'una o més d'aquestes llibreries serà imprescindible a l'hora d'extreure les característiques dels sons.

3.7.1 Marsyas

Marsyas (*Music Analysis, Retrieval and Synthesis for Audio Signals*) és una llibreria de codi obert per al processament d'àudio amb especial èmfasi amb les aplicacions relacionades amb el "*Music Information Retrieval*".

És una llibreria força potent, i que abasta totes les fases del processament del so, entre elles l'entrada i la sortida d'àudio usant diferents formats, per exemple l'mp3.

Referent a les característiques que és capaç d'extreure d'una mostra de so, es mostra bastant limitat.

URL	http://marsyas.info
Versió	0.3.6
Llenguatge	C++
Plataforma	Linux, Windows i Mac OSX
Llicència	GPL

3.7.2 CLAM

CLAM (*C++ Library for Audio and Music*) és un entorn de programació per a la recerca i desenvolupament d'aplicacions en el camp de la Música. Ofereix un entorn complet per a l'anàlisi, síntesi i processament de senyals.

Així com un paradigma particular per a gestionar fluxos de so a través de diferents actuadors, el qual incorpora fins i tot una interfície gràfica.

Ofereix una gran quantitat de característiques, fins a 72.

URL	http://clam-project.org/
Versió	1.4.0
Llenguatge	C++
Plataforma	Linux, Windows i Mac OSX
Llicència	GPL

3.7.3 Aubio

És una llibreria per a l'etiquetat d'àudio. Entre les seves possibilitats trobem la segmentació de sons en funció de cadascun dels seus atacs, detecció de tons, detecció del ritme, i la generació de melodies en format MIDI a partir de sons més complexes.

El seu nom prové d'àudio, amb un petit error tipogràfic, de manera anàloga als errors que solen haver-hi en els resultats de la llibreria

> *"The name aubio comes from 'audio' with a typo: several transcription errors are likely to be found in the results too."*

El que no ens ofereix gaire confiança a l'hora de fer-la servir.

Pel que fa a les característiques que és capaç de calcular, aquestes no són gaires, ja que és una llibreria que treballa a un nivell més alt, per exemple detectant el ritme d'un so o la seva melodia.

Cal dir que és molt usada en diversos projectes d'àudio força rellevants, com per exemple l'Audacity, el PureData o la llibreria CLAM, que hem vist anteriorment

URL	http://aubio.org
Versió	0.3.2
Llenguatge	C
Plataforma	Linux i Mac OSX
Llicència	GPL

3.7.4 LibXtract

LibXtract és una llibreria de baix nivell per a l'extracció de característiques musicals. Consta de 40 funcions, entre les quals trobem funcions auxiliars (per exemple per a calcular l'espectre d'una mostra de so) i de funcions per a extreure característiques.

URL	http://libxtract.sourceforge.net
Versió	0.6.2
Llenguatge	C
Plataforma	Linux i Mac OSX
Llicència	GPL

3.8 Algoritmes Implementats

Veiem ara alguns com funcionen alguns sistemes ja implementats amb èxit que hem implementat.

Hem triat aquests algoritmes perquè representen les diferents alternatives que existeixen avui en dia en l'àudio fingerprinting. En primer lloc el basat en característiques, que intenta extreure característiques d'alt nivell del so. En segon lloc, el basat en la DCT, que és força novedós i prometedor. I finalment i en tercer lloc, un que combina el millor dels dos anteriors.

3.8.1 Basat en característiques

Per a cada segment es calculen les característiques anteriorment vistes a l'apartat 3.6.

Un cop fet això, es crea el fingerprint, que consta d'un vector multidimensional, on cada dimensió representa el valor d'una característica.

Finalment, afegim el fingerprint juntament amb les metadades relacionades amb el segment en qüestió (per exemple el nom de l'artista i el títol de la cançó a la qual pertany) a la base de dades.

Un cop fet això, ja podem calcular el fingerprint d'altres segments, per d'aquesta manera, cercar el fingerprint més similar al catàleg, i finalment obtenir les metadades associades, per d'aquesta manera, tenir més informació sobre la mostra d'àudio.

3.8.2 Basat en la DCT

Aquest algoritme va ser creat amb la finalitat de ser força robust a les distorsions, i es caracteritza per no fer servir cap de les característiques vistes anteriorment. En comptes d'això, crea múltiples cadenes de hash, que són generades a partir de la transformada discreta del cosinus, de la seqüència temporal d'energies de diferents subbandes de l'espectre.

En primer lloc subdividim el segment en 32 subsegments, un cop fet això, calculem l'espectre de cada subsegment, i ho subdividim en diferents bandes de freqüències.

Llavors per a cada banda de freqüències, calculem l'energia total, sumant l'amplada de totes les freqüències que la formen. Un cop fet això, calculem la DCT de les energies de les diferents subbandes.

Sigui CK(n.m) el k-èssim coeficient, de la m-èssima subbanda del n-èssim segment, i L el número de de subbandes calculem:

$ED_k(n,m) = C_k(n, m) - C_k(n, m+1) - C_k(n-L, m) - C_k(n-L, m+1)$

on ED_k representa la k-èssima diferència de la subbanda m en el subsegment n.

Llavors sigui,

$$F_k(n,m) = \begin{cases} 1, ED_k(n,m) > 0 \\ 0, ED_k(n,m) < 0 \end{cases}$$

generem el fingerprint amb

$$F_k(n) = \left[F_k(n,0), F_k(n,1) \ldots F_k(n,31) \right]$$

3.8.3 Basat en l'evolució de les característiques

Aquest algoritme pretén codificar l'evolució de cada característica al llarg del segment. En primer lloc dividirem la mostra de so en diferents segments. Un cop fet això extraurem les característiques que millor resultat ofereixin a l'hora de discriminar sons. Per a finalment, usar la DCT per a codificar l'evolució de cada característica al llarg dels diferents segments.

4 Implementació

En aquest capítol exposarem en profunditat els passos a seguir per a realitzar una aplicació que ens permeti fer la comparativa entre algoritmes.

4.1 Requeriments Funcionals

4.1.1 Requeriments de les dades d'entrada i sortida

Els arxius d'entrada estaran codificats en mp3 tindran el format “Artista – Títol.mp3”, on Artista i Títol podran estar formats per diferents paraules, separades exclusivament per espais, tampoc es permeten caràcters ASCII extès, és a dir, ni caràcters estranys ni accents.

El programa tindrà dos paràmetres on especificarem:

- El directori on es troben els arxius de so que serviran per a crear el catàleg
- El directori on es troben els arxius que volem cercar al catàleg

4.1.2 Requeriments de funcionalitat

El programa haurà de generar els fingerprints per als arxius del directori de segments i guardar aquests fingerprints, juntament amb el nom i l'artista de cada segment.

Posteriorment per a cada arxiu del directori de test, es cercarà al catàleg, usant cadascun dels algoritmes (basat en característiques, basat en la DCT i basat en l'evolució de les característiques) quin és el segment més similar, i si resulta que és el mateix, apuntarem un encert.

Tant el procés de càlcul del fingerprint, com el de cerca d'elements al catàleg estaran mesurats respecte al temps que triguen.

Finalment el programa donarà les taxes d'encert de cadascun dels algoritmes, així com

mitjanes del temps que triga cadascun dels algoritmes tant en la generació del fingerprint com en realitzar la cerca al catàleg.

4.2 Llenguatges i entorn

En primer lloc hem decidit treballar amb llenguatge C. Aquesta decisió l'hem presa tenint en compte, que la finalitat d'aquest projecte és comprovar l'eficiència dels algoritmes, i per tant ens interessa un llenguatge d'un relatiu baix nivell, i que ens permeti un accés a les dades eficaç.

Un problema d'aquest tipus de llenguatges és la poca orientació a objectes que permet, per la qual cosa l'estructura del codi pot semblar una mica complicada.

El programa s'ha creat amb l'entorn de desenvolupament integrat Eclipse©, juntament amb el connector per a C/C++.

Les llibreries no habituals utilitzades han estat:

- libfftw3 : Per a realitzar les transformades del cosinus i de Fourier
- libXtract : Per a extreure les caracteristiques dels sons
- glu.h/glut.h : Per a mostrar gràficament senyals i espectres
- types.h/dirent.h : Per a llistar i recórrer directoris
- sndfile.h / lame.h : Per a llegir i escriure arxius wav i mp3
- fcntl.h, sys/ioctl.h, sys/soundcard.h : Per a reproduir arxius
-

Finalment hem decidit utilitzar la libXtract, principalemt per descart de les altres, la Marsyas perquè no calculava prou característiques, la CLAM perquè va molt més enllà del càlcul de característiques, i la Aubio perquè a part de la seva poca fiabilitat les característiques amb les que treballa solen trobar-se en un nivell massa elevat.

Alhora la libXtract, és força potent, i ofereix un bon conjunt de característiques.

4.3 Mòduls del sistema

Veiem a continuació de quines parts consta el nostre sistema. En primer lloc veurem el preprocessament, que s'encarrega de preparar els arxius musicals per tal de segmentar-los i donar-los el format adequat.

Posteriorment veurem en detall el funcionament dels algoritmes d'àudio fingerprinting implementats.

Finalment veurem alguns mòduls auxiliars que han estat d'ajuda al llarg del procés de creació de tot el sistema.

4.3.1 Preprocessament

4.3.1.1 Segmentador

Aquest és l'encarregat de llegir els arxius mp3 que contenen les cançons, i generar, a partir de les característiques del so, els diferents segments.

Inicialment es van realitzar diverses proves per a construir un algoritme relativament senzill tenint en compte la gran dificultat d'aquest pas, el fet que ens hem de basar en l'hora de fer la segmentació, és a dir, una cançó ha d'estar igualment dividida independentment del tros que tinguem disponible.

Després de diversos desenvolupaments candidats, es va veure que l'abast del problema excedia els objectius del projecte, i es va decidir utilitzar una solució ja implementada anomenada *Music Audio Tempo Estimation and Beat Tracking*.

Com indica el seu nom, aquest algoritme, en primer lloc detecta els cops (beats) de les cançons, i a partir d'aquests, calcula un tempo candidat, de manera que els beats es produeixin d'acord amb el tempo.

Un cop tenim els frames que estableixen la separació entre segments, creem un arxiu per a

cada segment. Aquests arxius són els que farem servir com a entrada de la nostra aplicació.

Aproximadament, cada cançó genera uns 800 segments, per tant, per a poder abastar un ventall més ampli d'estils musicals, en seleccionarem 50. Altrament o bé tenim un catàleg excessivament petit, o el cost computacional (principalment en memòria) que representaria faria la comparativa inassumible.

Consta d'un script realitzat en Matlab© que llegeix els arxius mp3, i els transforma en diversos segments. Aquest segments, són posteriorment codificats en arxius mp3 amb un bitrate de 32kbps, un samplerate de 44100 hz, i un sol canal.

Els paràmetres que li passem són el directori amb les cançons inicials, i el directori on volem que guardi els segments.

4.3.1.2 Formatador

En aquest apartat ens assegurem que els arxius d'entrada consten només d'un canal, és a dir, estan en mono. En cas contrari els transformem.

Addicionalment, i de cara a intentar millorar el rendiment en cost computacional de l'aplicació , es podria reduir tant el bitrate com el samplerate de les mostres, però com que el nostre objectiu és comparar els algoritmes ens hem estalviat aquest pas, ja que no afecta a la comparació.

4.3.2 Algoritmes implementats

En primer lloc, hem implementat l'algoritme més elemental, el basat en característiques. Davant la poca concreció que exposa la literatura del tema sobre quines són les més òptimes, hem decidit crear un algoritme que faci ús de totes les possibles característiques, per a posteriorment avaluar-les.

En segon lloc, s'ha implementat un dels algoritmes més innovadors, el qual, segons el seu autor, ofereix millors resultats. Es tracta del basat en la DCT, que a més, té la particularitat de tenir en compte el funcionament del sistema auditiu humà.

Finalment, i a mode de prova, hem seleccionat les característiques que ofereixin millors resultats del primer algoritme, i hem creat el fingerprint d'una manera lleugerament diferent,intentant modelar l'evolució d'aquestes característiques al llarg del so. En un moment de brillantor, hem decidit anomenar aquest algoritme Basat en l'evolució temporal de les característiques.

4.3.2.1 Basat en característiques

Aquest algoritme en primer lloc calcula l'espectre de potència del so així com l'espectre de pics, que posteriorment seràn útils per a crear el fingerprint a partir de les característiques explicades a l'apartat 3.6 (mitjana, variança, desviació estàndard, desviació mitjana, asimetria, curtosi, mitjana quadràtica (RMS), mitjana espectral, variança espectral, desviació estàndard espectral, desviació mitjana espectral, asimetria espectral, curtosi espectral, centroide, irregularitat, dispersió de l'espectre, triestímuls, suavitat de l'espectre (Smoothness), propagació de l'espectre (spectral spread), rolloff, planesa de l'espectre (Flatness), pendent de l'espectre, inharmonia espectral, rati entre parells i senars, volum (loudness) i rati de creuaments amb zero.)

El fingerprint d'aquest algoritme està format per un vector de 22 elements (un per cada característica), així com nom de l'artista i el títol de la cançó d'on ha estat extret.

Un cop s'han creat els fingerprints de tots els segments, normalitzem el valor de totes les característiques, en funció del valors màxims i mínims que pren cadascuna d'elles.

A l'hora de realitzar la cerca, calculem el fingerprint de la mostra d'entrada, normalitzem el valor de les seves característiques, i calculem la distància euclidiana entre el seu fingerprint i el de la resta d'elements de la base de dades.

Aquell en què la distància sigui menor, i no sigui superior a un cert llindar, ho donem com a resultat.

4.3.2.2 Basat en la DCT

En el mètode basat en la DCT en primer lloc dividim la mostra de so d'entrada en 32 frames de la mateixa mida.

Un cop fet això, calculem l'espectre de potència del so. Ara, per cadascuna de les subbandes logarítmicament equiespaiades de l'escala de Bark, en calculem la DCT.

Els coeficients que farem servir com a separadors de bandes són els següents:

400, 510, 630, 770, 920, 1080, 1270, 1480, 1720, 2000, 2320, 2700, 3150, 3700, 4400, 5300, 6400, 7700, 9500, 12000, 15500, 20000

Un cop ja tenim els coeficients per a cada banda i per a cada subsegment, és el moment de dispersar la informació dels diferents coeficients, d'aquesta manera aconseguim dispersar les característiques de cada subbanda entre els seus veïns.

Un cop fet això, tornem a aplicar la DCT al conjunt de valors que pren cada coeficient de cada subbanda al llarg dels diferents subsegments.

Finalment, cada fingerprint consta d'una matriu de valors que té 22 files, per 10 columnes, on a cada fila hi ha els coeficients de Fourier que modelen el comportament de cada característica.

Per a trobar l'element més semblant a un so donat, calcularem el fingerprint del so, i posteriorment compararem aquest nou fingerprint amb cadascun dels de la base de dades mitjançant la distància euclidiana (la distància en valor absolut entre cadascun dels seus elements).

Aquell en què la distància sigui menor, i no sigui superior a un cert llindar, ho donem com a resultat.

4.3.2.3 Basat en l'evolució temporal de les característiques

Aquest algoritme, pretén aprofitar el millor dels dos anteriors. Del primer, fem servir les característiques que han resultat més eficients a l'hora de discriminar sons, i del segon, farem servir la potència de compressió de la informació que ens ofereix la DCT.

A més a més, cal dir que aquest algoritme no codifica el so en sí, sinó que codifica exclusivament la seva evolució. Això hauria de mostrar-se com a força determinant a l'hora de intentar recuperar sons amb una quantitat alta de soroll.

Per a triar les característiques que més ens ajuden a discriminar els sons, hem modificat el primer algoritme per tal de que faci servir tan sols una característica per a identificar quin dels fingerprints és el més semblant, d'aquesta manera, podem veure quines són les característiques més útils.

Els resultats d'aquesta prova són els següents:

Spectral Inharmonicity	23,30	21,02	19,13	18,58	20,5074
Tristimuls 1	20,08	20,64	19,51	18,20	19,6066
Tristimuls 2	14,20	14,02	14,96	14,75	14,4832
Spectral Flatness	15,34	14,02	12,50	12,64	13,6250
Flatness db	15,34	14,02	12,50	12,64	13,6250
Tonality	15,34	14,02	12,50	12,64	13,6250
Tristimuls 3	10,61	13,26	9,28	13,41	11,6385
Standard Deviation	14,39	10,98	9,28	10,34	11,2510
RMS Amplitude	14,02	10,04	8,33	9,20	10,3955
Irregularity j	10,42	8,90	9,09	9,20	9,4011
Skewness	7,58	11,36	9,85	6,32	8,7774
Odd Even Ratio	7,95	7,39	9,09	7,28	7,9279
Power	7,95	7,39	9,09	7,28	7,9279
Kurtosi	7,39	8,90	8,52	6,13	7,7352
Spectral Spread	6,06	7,95	7,58	8,24	7,4571
Crest	6,44	8,52	7,58	7,09	7,4065
Irregularity k	7,58	6,82	6,82	6,90	7,0272
Centroid	5,87	7,58	8,14	6,51	7,0261
Spectral Mean	6,06	7,01	7,77	6,71	6,8846
Rolloff	4,73	7,58	7,95	5,75	6,5031
Zero Crossing Rate	6,82	6,82	3,41	5,75	5,6982
Mean	6,25	4,17	2,46	4,41	4,3212
Variance	6,25	4,17	2,46	4,41	4,3212
Kurtosi Espectral	3,22	4,92	5,11	3,64	4,2243
Average Deviation	6,63	3,98	1,70	3,83	4,0355
Espectral Variance	3,03	4,73	3,22	2,87	3,4646
Espectral Standard Deviat	3,03	4,73	3,22	2,87	3,4646
Skewness Espectral	3,41	2,65	1,33	2,68	2,5171
Loudness	1,14	2,27	2,27	1,15	1,7078

Taula 1: Utilitat de les característiques per a discriminar sons

per tant les característiques amb les que ens quedem son :

L'inharmonia espectral, els tres triestímuls, la suavitat de l'espectre, la suavitat de l'ona la tonalitat, la desviació estàndard, la mitjana quadràtica (RMS) i la irregularitat.

Per a calcular el fingerprint, en primer lloc dividim cada segment en 32 subsegments equiespaiats.

Un cop fet això, calculem el fingerprint del model basat en característiques de cadascun d'aquests subsegments.

Finalment modelarem l'evolució de cadascuna d'aquestes característiques al llarg dels diferents segments utilitzant, una vegada més, la DCT.

El fingerprint final, serà una matriu de nombres en punt flotant amb tantes files com característiques haguem modelat, i amb tantes columnes com coeficients de la DCT haguem decidit guardar. En el nostre cas 10.

A part d'això, l'estructura que guardarem contindrà també el nom de l'intèrpret de la cançó, així com el títol del tema al qual pertany el segment.

Per a trobar el fingerprint més semblant, d'una mostra de so, calculem el fingerprint de la mostra d'entrada, i el comparem mitjançant la distància euclidiana amb la resta de fingerprints de la base de dades.

Aquell que tingui una menor distància, i aquesta sigui inferior a un cert llindar, ho donarem com a resultat.

4.3.3 Mòduls auxiliars

S'han creat alguns mòduls que ajuden a la tasca de visualitzar, enregistrar i escoltar diferents senyals.

4.3.3.1 Lector i enregistrador d'àudio

En primer lloc el que hem hagut de fer han estat funcions, que ens permetin llegir arxius mp3. Ho hem fet amb la llibreria LAME pels arxius en format mp3, i la llibreria sndfile per els de format wav.

També hem creat un enregistrador de so en format mp3, per ajudar-nos al desenvolupament, especialment en l'etapa de preprocés.

4.3.3.2 Visualitzador

Addicionalment, s'ha creat una petita interfície en OpenGL, que permet visualitzar en mode gràfic la forma dels senyals de so, així com els seus espectres.

4.3.3.3 Reproductor

Permet escoltar arxius un cop processats, per a verificar la funció de preprocessament, i també per a conèixer si hi ha semblances en el cas de que es faci una identificació errònia.

4.3.3.4 Enregistrador de text

Permet guardar en mode text arxius de so així com espectres, per a la seva posterior visualització amb programes externs.

4.4 Entorn de testeig

4.4.1 Bateria de proves

Ens interessa tenir cançons de diferents estils musicals, per a veure com es comporta cada algoritme. Per a aconseguir aquest catàleg musical o bateria de proves vam buscar diverses fonts públiques que permetessin la baixada legal de música. Entre totes elles vam escollir la disponible a mp3.com i hem creat un script per a recuperar totes les cançons disponibles.

Està escrit en *Python* i usa la llibreria urrlib2 per a obtenir tant la pàgina web com els arxius mp3, i la llibreria *Beautiful Soup* per parsejar l'html.

.De seguida ens hem adonat, que la quantitat d'arxius dins d'aquesta pàgina és inabarcable, ja que hi ha prop de 30.000 cançons. A més a més, moltes d'elles són de molt baixa qualitat, i hi predominen estils com l'electrònica, cosa que fa que no sigui gaire adequat per a ser usat com a conjunt de test per al nostre programa.

Per a solucionar-ho, hem seleccionat un conjunt de cançons, 70 en total, intentant quedar-nos només amb aquelles que han tingut un procés de gravació mínimament professional, i que representin una mostra equitativa dels estils més escoltats actualment (Rock, Metal, Electrònica, Cançó d'autor, Rumba i Folk).

A partir d'aquí, hem creat quatre conjunts de test, formats per 500 segments per a crear el catàleg, i 90 per a posar-lo a prova.

4.4.2 Tipus de sorolls

Ens interessa tenir una base de dades de sorolls per a simular l'efecte que provoca la transmissió del so a través de diversos mitjans, principalment l'aire.

Això ens permetrà comprovar l'efectivitat que tenen els algoritmes en condicions habituals.

S'han fet servir cinc tipus diferents de soroll, tots força comuns al món real.

Blanc

Aquest soroll, anomenat “blanc” es correspon a un senyal on tots els components freqüencials tenen la mateixa amplada. És el soroll que solem escoltar en una ràdio sense tenir cap emissora sintonitzada.

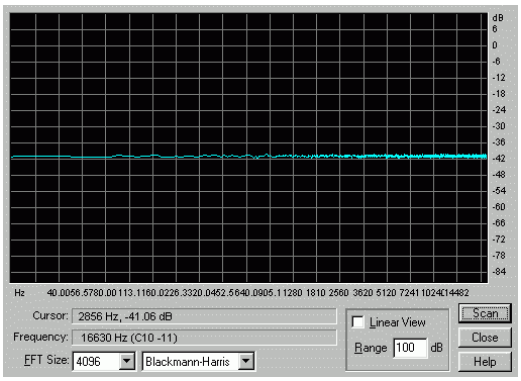


Figura 4.4.1: Espectre del soroll Blanc

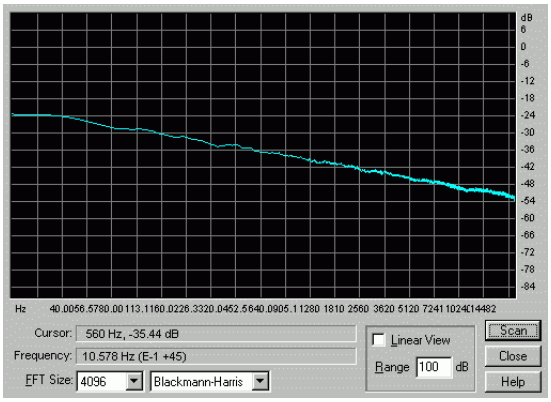


Figura 4.4.2: Espectre del soroll Rosa

Rosa

Aquest és el soroll té una distribució de freqüència 1/f, i representa un augment de 3 dB per octava del rolloff. És conegut per ser el més present en entorns naturals.

Marró

Aquest soroll té una distribució de freqüències de 1/(f^2), i per tant, un increment de potència de 6 dB.

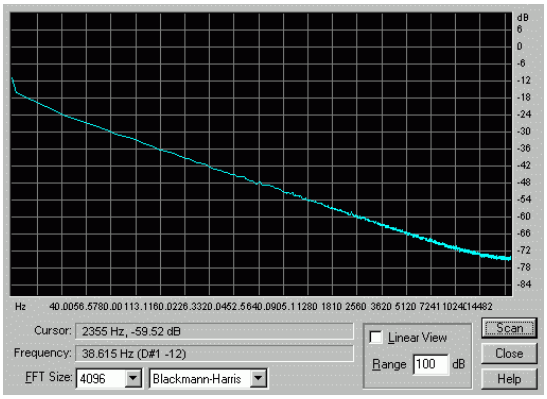


Figura 4.4.3: Espectre del soroll Marró

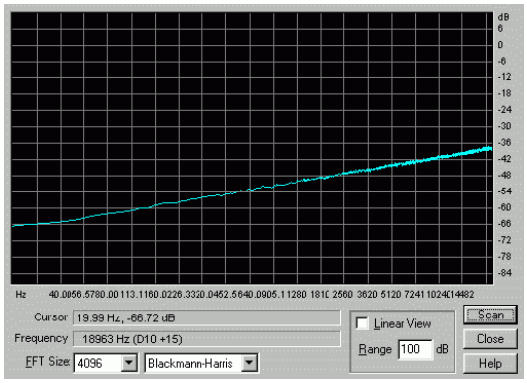


Figura 4.4.4: Espectre del soroll Blau

Blau

Aquest so el podem considerar com l'invers del rosa, ja que la seva distribució de freqüències és proporcional a f. Representa 3 dB.

Violeta

Finalment el violeta és l'oposat del marró, representa 6 dB i la seva distribució de freqüències és de f^2.

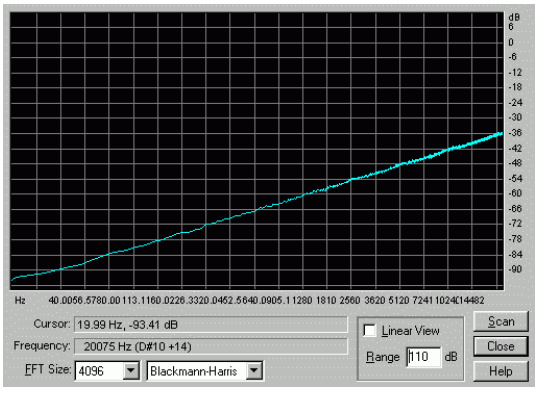


Figura 4.4.5: Espectre del soroll Violeta

4.4.3 Mesures

L'objectiu principal del projecte, és determinar quin és l'algoritme de fingerprint més efectiu a l'hora de discriminar sons. Per tant, la taxa d'encerts de cadascun d'ells és la mesura principal del seu èxit.

Hem provat la taxa d'encert de cadascun dels algoritmes, en diferents condicions de soroll, és a dir, sense soroll, i amb cadascun dels sorolls vistos anteriorment.

També hem avaluat quin és el temps que triga cada algoritme en generar el fingerprint, així com en trobar el més similar a un fingerprint d'entrada.

5 Resultats i conclusions

5.1 Característiques

Com hem vist a l'apartat 4.3.2.3 les característiques que han resultat més efectives a l'hora de discriminar sons han estat l'inharmonia espectral, els tres triestímuls, la suavitat de l'espectre, la suavitat de l'ona, la tonalitat, la desviació estàndard, la mitjana quadràtica (RMS) i la irregularitat.

Era bastant esperat, ja que totes elles són d'alt nivell, excepte la desviació estàndard, que sobta una mica, però creiem que realment és una bona característica, ja que resulta força rellevant en tots 4 tests.

Addicionalment, la majoria d'aquestes són també utilitzades en els algoritmes basats en característiques explicats a l'apartat 3.4.

5.2 Temps requerit per a cada algoritme

Creació FP

	Test1	Test2	Test3	Test4
Algoritme 1	1152	1148	1150	1227
Algoritme 2	6334	6222	6219	7281
Algoritme 3	2757	2711	2715	2871

Cerca FP

	Test1	Test2	Test3	Test4
Algoritme 1	5	4	4	4
Algoritme 2	56	18	57	95
Algoritme 3	8778	8488	8570	9225

	Creació FP	Cerca FP
Algoritme 1	1169,25	4,25
Algoritme 2	6514	56,5
Algoritme 3	2763,5	8765,25

Taula 2: Resultats del temps empleat en calcular i cercar fingerprints

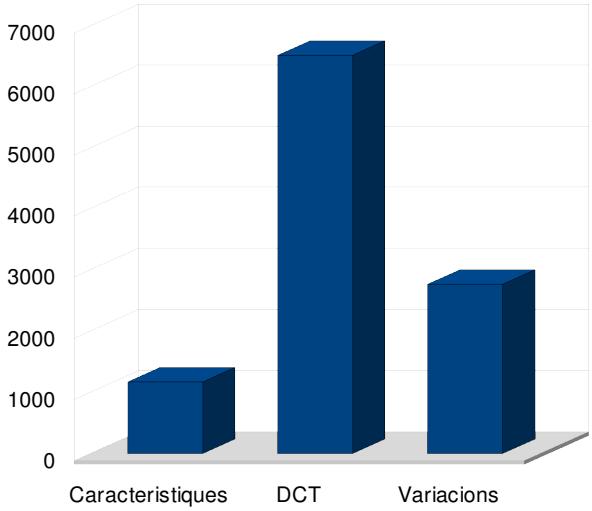


Figura 5.2.1: Comparativa del temps de creació de cada fingerprint

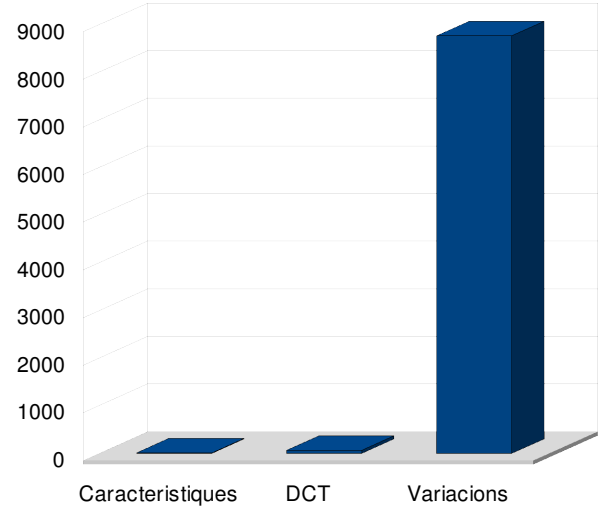


Figura 5.2.2: Comparativa del temps de cerca de cada algoritme

Veiem com el temps de càlcul del fingerprint del tercer algoritme és lleugerament superior al del primer, tot i que calcula diverses característiques fins a 32 vegades, però també cal recordar que el numero de dades amb què ho fa és molt inferior.

Sorprèn també el temps requerit per el segon algoritme, a l'hora de calcular el fingerprint, però si tenim en compte que fem aproximadament que per cada segment calculem 32 espectres i 33 DCT's, potser resulta fins i tot ràpid.

Respecte al temps de cerca empleat, veiem que en el primer i segon algoritmes és realment ràpid, mentre que en el tercer no ho és gens. Això és degut a que en el primer i el segon comparem vectors, mentre que en el tercer hem de comparar matrius.

5.3 Taxa d'encerts dels algoritmes

TEST 1

	Sense Soroll	Soroll Blanc	Soroll Rosa	Soroll Marró	Soroll Blau	Soroll Violeta
Algoritme 1	100,00	31,03	28,74	24,14	50,57	50,57
Algoritme 2	98,85	86,21	85,06	87,36	98,85	98,85
Algoritme 3	94,25	56,32	29,89	80,46	59,77	45,98

TEST 2

	Sense Soroll	Soroll Blanc	Soroll Rosa	Soroll Marró	Soroll Blau	Soroll Violeta
Algoritme 1	100,00	27,27	22,73	19,32	34,09	32,95
Algoritme 2	100,00	86,21	85,06	87,36	98,85	98,85
Algoritme 3	100,00	72,73	35,23	88,64	68,18	51,14

TEST 3

	Sense Soroll	Soroll Blanc	Soroll Rosa	Soroll Marró	Soroll Blau	Soroll Violeta
Algoritme 1	100,00	21,59	23,86	29,55	27,27	28,41
Algoritme 2	98,86	90,91	88,64	88,64	97,73	98,86
Algoritme 3	100,00	64,77	36,36	78,41	61,36	46,59

TEST 4

	Sense Soroll	Soroll Blanc	Soroll Rosa	Soroll Marró	Soroll Blau	Soroll Violeta
Algoritme 1	100,00	27,27	25,00	23,86	40,91	43,18
Algoritme 2	97,73	92,05	89,77	88,64	97,73	96,59
Algoritme 3	100,00	54,55	35,23	79,55	54,55	40,91

	Sense Soroll	Soroll Blanc	Soroll Rosa	Soroll Marró	Soroll Blau	Soroll Violeta
Caracteristique	100	25,37225	24,797525	25,636775	34,80275	35,08685
DCT	99,14445	88,558	86,84695	87,99635	98,28895	98,8571
Variacions	98,5625	64,6475	34,46	81,48	62,6675	47,575

Taula 3: Comparativa de les taxes d'encert dels algoritmes

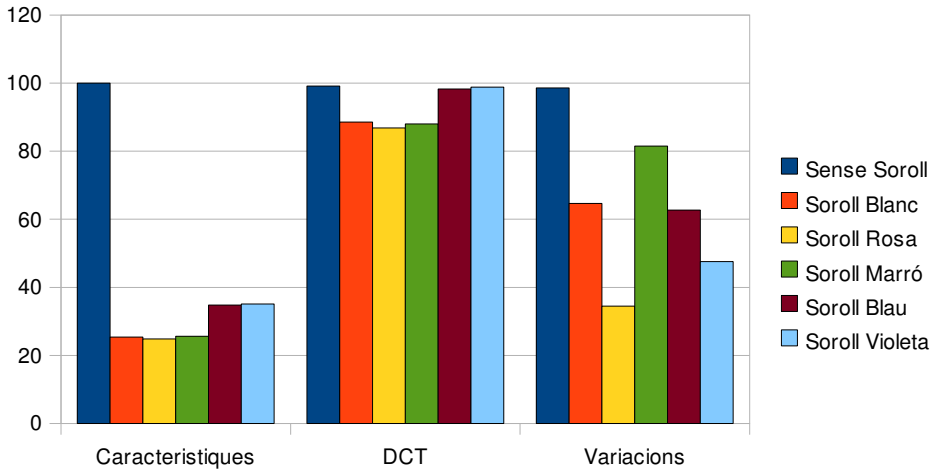


Figura 5.3.1: Comparativa dels diferents algoritmes en funció del soroll

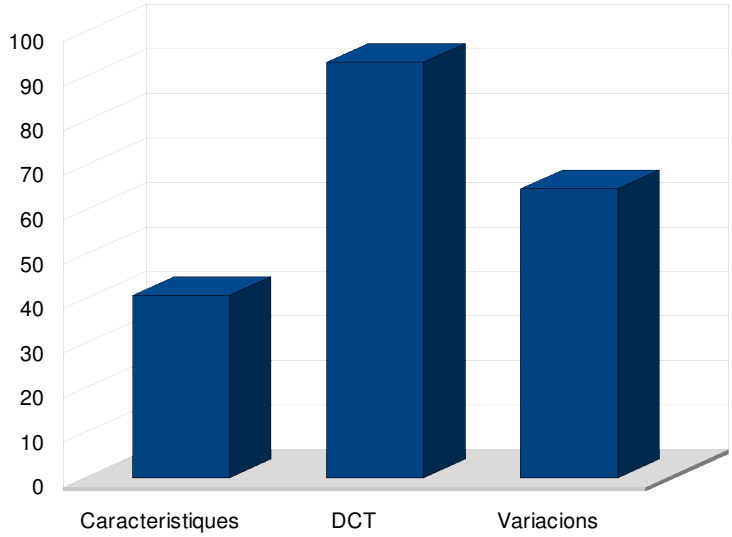


Figura 5.3.2: Comparativa general dels tres algoritmes

Com hem vist, l'algoritme que aconsegueix una major taxa d'encerts tenint en compte tots els tipus de sorolls és el basat en la DCT.

El que confirma, que tot i que les característiques dels sons, poden ser molt útils a l'hora de discriminar-los, encara no són suficients per a descriure'ls inequívocament.

També veiem com l'algoritme basat en l'evolució de les característiques aconsegueix una taxa d'encerts força superior al que codifica el valor de les característiques, el que confirma que no és tan important el valor absolut de les característiques, sinó la seva evolució temporal.

5.4 Conclusions finals

Hem construït un sistema que permet comparar diferents algoritmes de fingerprinting. També hem descobert que l'algoritme més efectiu és el basat en la DCT, el qual té una taxa d'encerts força elevada, no és dels més costos computacionalment, i és molt robust al soroll.

A més a més, tots els algoritmes han aconseguit taxes d'encert raonables, i sense soroll, tots tenen un encert superior al 98%. Per tant, podem dir que s'han aconseguit els objectius inicials del projecte.

6 Ampliacions i millores

Una de les possibles millores seria l'ampliació de la mida del catàleg, ja que actualment és força reduïda. Per a fer-ho, s'hauria de guardar el catàleg al disc dur, i no en memòria com es fa actualment, altrament, la memòria s'exhaureix de seguida.

També seria força interessant, afegir al catàleg versions lleugerament diferents de cançons, com per exemple versions enregistrades per altres grups, o versions en directe de cançons que abans només teníem en enregistrades en estudi.

Una altra possible millora de cara a augmentar el rendiment de l'aplicació, seria reutilitzar càlculs que són comuns a alguns algoritmes (per exemple el càlcul de l'espectre), ja que actualment aquests es calculen per a cada algoritme.

7 Bibliografia

[1] J. S. Downie, ‘‘Music information retrieval,’’ Annu. Rev. Inf. Sci. Technol., 37, 295–340 (2003).

[2] N. Orio, ‘‘Music information retrieval: A tutorial and review,’’ Found. Trends Inf. Retr., 1, 1-90 (2006)

[3] Haitsma, J. and Kalker, T. (2002b). A highly robust audio fingerprinting system. In Proceedings of the International Symposium on Music Information Retrieval, Paris, France.

[4] Batlle, E., Masip, J., and Guaus, E. (2002). Automatic song identification in noisy broadcast audio. In Proc. of the SIP.

[5]Kirovski, D. and Attias, H. (2002). Beat-id: Identifying music via beat analysis. In 5th IEEE Int. Workshop on Multimedia Signal Processing: special session on Media Recognition, US Virgin Islands, USA.

[6] Tzanetakis, G. and Cook, P. (1999). Multifeature Audio Segmentation for Browsing and Annotation. Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.

[7] C. Burges, J. Platt, and S. Jana, "Extracting Noise-Robust Features from Audio Data," in Proc. of the ICASSP, Florida, USA, May 2002.

[8] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison Wesley, 1999.

[9] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In Workshop on Multimedia Signal Processing, 2002.

[10] M. F. McKinney and J. Breebaart. Features for audio and music classification. In Proc ISMIR, 2003.

[11] Hwan Sik Yun1 and Nam Soo Kim1, "Audio fingerprinting based on multiple hashing in DCT domain" in IEEE Signal Processing Letters, 2009

[12] G. Tzanetakis, P.Cook “Multi-feature AudioSegmentation for Browsing and Annotation”,Proc.IEEE Workshop on Appl. Signal Proc.. toAudio and Acoustics(WASPAA),1999

[13] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project”, CUIDADO I.S.T. Project Report, 2004.

[14] D. Rocchesso "Introduction To Digital Audio Signal Processing" (2005)

8 Resums

8.1 Resum

L'evolució ens els últims decennis de les possibilitats relacionades amb les tecnologies de la informació han provocat l'aparició de diferents camps, entre ells l'anomenat “recuperació de música basant-se en el contingut”, que tracta de calcular la similitud entre diferents sons.

En aquest projecte hem fet una recerca sobre els diferents mètodes que existeixen avui en dia, i posteriorment n'hem comparat tres, un basat en característiques del so, un basat en la transformada discreta del cosinus, i un que combina els dos anteriors.

Els resultats han mostrat, que el basat en la transformada de Fourier és el més fiable.

8.2 Resumen

La evolución en las últimas décadas de las posibilidades relacionadas con las tecnologías de la información han resultado en la aparición de diferentes campos, entre ellos el llamado 'Recuperación de música basándose en el contenido', el cuál trata de calcular la simlitud entre diferentes sonidos.

En este proyecto hemos echo una investigación sobre los diferentes métodos que existen hoy en día, y posteriormente hemos comparado tres de ellos, uno basado en las características del sonido, otro basado en la transformada discreta del cosinus, y uno que combina los dos anteriores.

Los resultados han demostrado que el basado en la transformada discreta del cosinus es el que ofrece mejores resultados.

8.3 Abstract

The evolution in the las decades of the possibilites related to the information technologies have led to the emergence of different fields, among those, the so-called 'Content-based music retrieval', which are focused on calculating the similarity between different sounds.

In this project we have conducted a research about the different methods that exists nowadays and compared three of them, one based on the characteristics of sound, one based on discrete cosinus transform, and one that combines both.

The results have shown that the discrete consinus transform based is which offers better performance.

Índex de figures

Índex de taules